



**POLITECHNIKA
BYDGOSKA**

Wydział Telekomunikacji,
Informatyki i Elektrotechniki



DOKUMENTACJA PROJEKTU

BAZA DANYCH FIRMY KURIERSKIEJ - MYSQL

JULIA KAŁUŻA

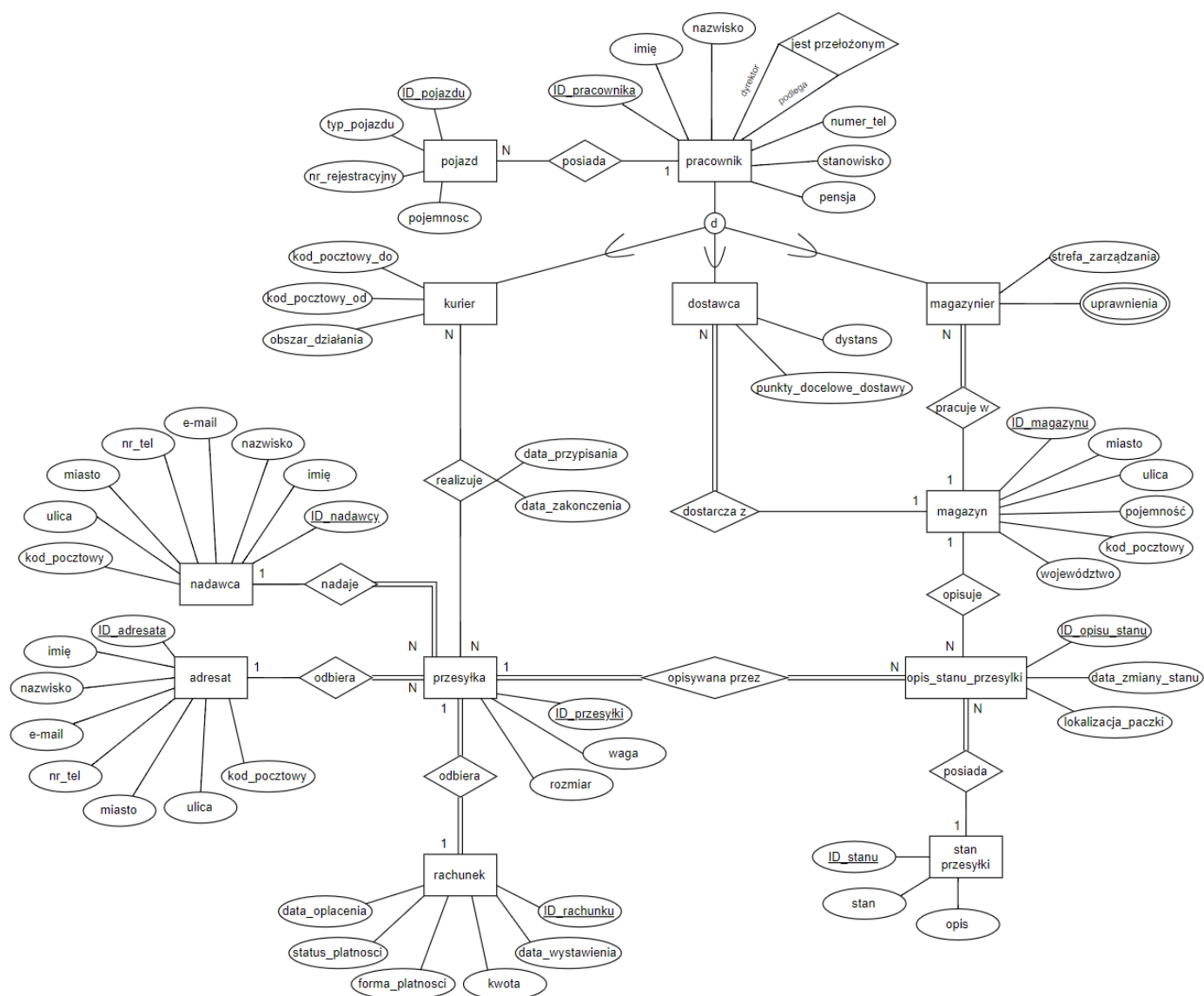
INFORMATYKA STOSOWANA
GRUPA 2, SEM. 3, 2024/2025

Spis treści

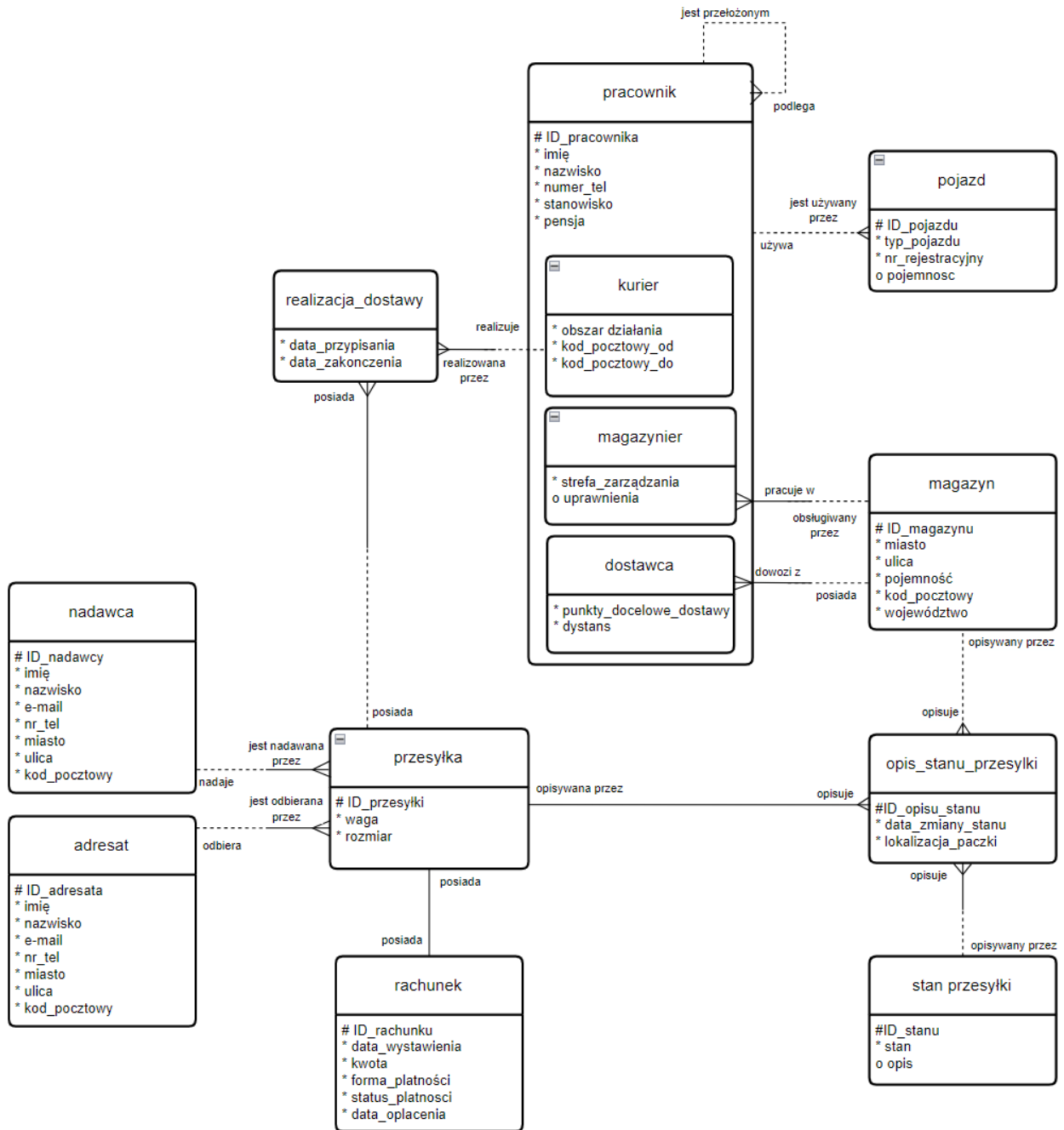
Diagramy	3
Notacja Chena	3
Notacja Barkera	4
Notacja UML	5
Model relacyjny	0
Opis modeli relacyjnych.....	7
Tworzenie tabel i wypełnianie danymi	11
a) Adresat	11
b) Dostawca	12
c) Kurier	14
d) Magazyn	15
e) Magazynier	16
f) Nadawca	17
g) Opis stanu przesyłki	18
h) Pojazd	22
i) Pracownik	23
j) Przesyłka	26
k) Rachunek	27
l) Realizacja dostawy	28
m) Stan przesyłki	29
n) Uprawnienia	30
Zapytania SQL.....	31
a) Dopisywanie danych do bazy – INSERT INTO	31
b) Modyfikowanie danych w bazie – UPDATE	32
c) Usuwanie danych z bazy – DELETE	32
d) Pobieranie danych z bazy – SELECT	33
1) SELECT – wyświetlenie danych	33
2) Złączenie wewnętrzne – INNER JOIN	33
3) Złączenie zewnętrzne – LEFT OUTER JOIN.....	34
4) Złączenie zewnętrzne - RIGHT OUTER JOIN.....	35
5) Grupowanie – GROUP BY	36

6) Sortowanie – ORDER BY	36
7) Podzapytanie nieskorelowane	37
8) Podzapytanie skorelowane	37
9) Klauzula HAVING	38
10) Klauzula LIKE	38
11) Klauzula IN	38
12) Klauzula ANY	39
13) Klauzula ALL	39
14) Klauzula EXISTS	40
15) Dodatkowe zapytanie	40
Procedury składowane	41
Wyzwalacze	44
Aplikacja.....	48
Połączenie z bazą.....	48
Tworzenie raportu w pythonie – ReportLab	48
Działanie aplikacji	50
Szablony raportów	52

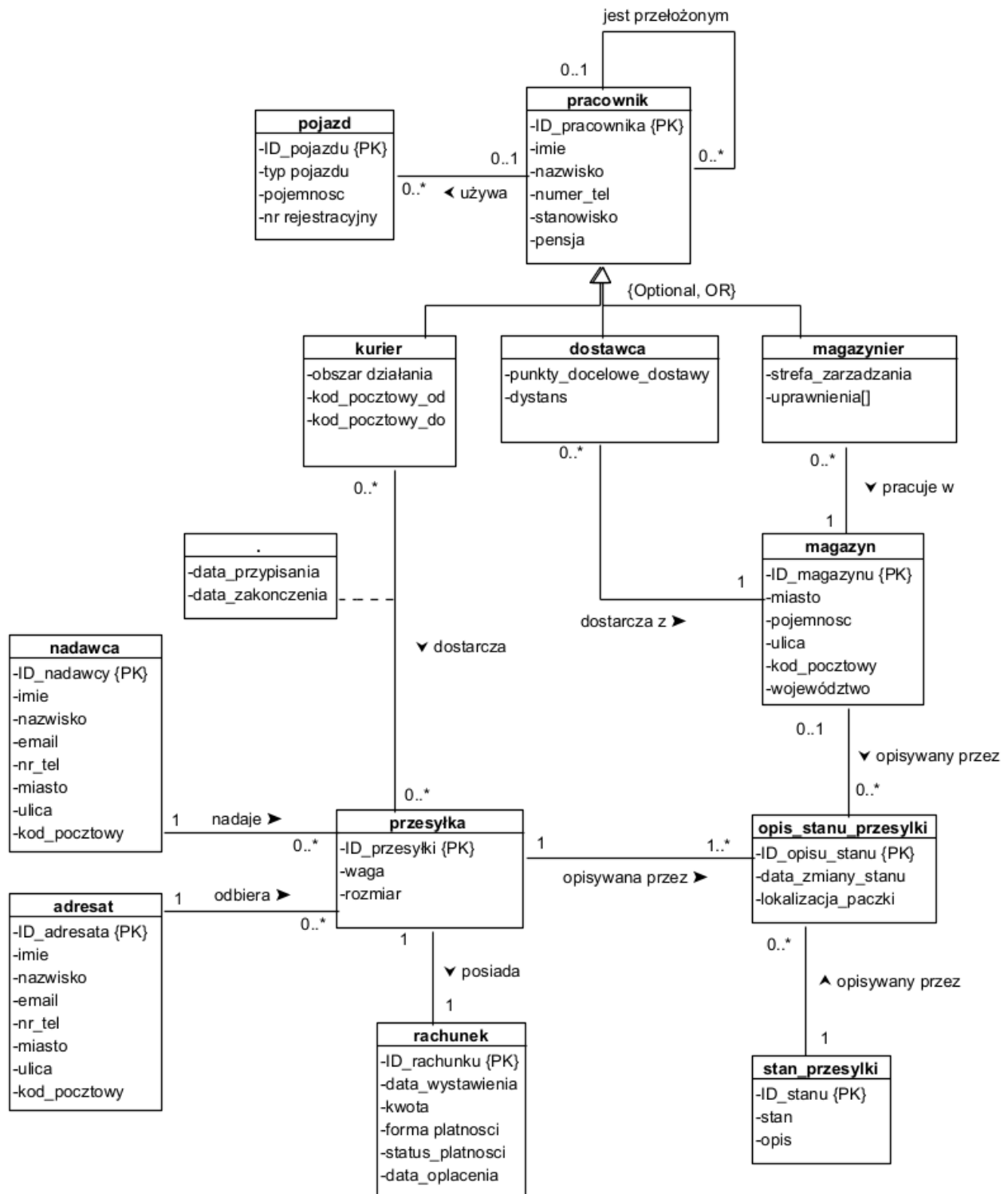
Notacja Chena



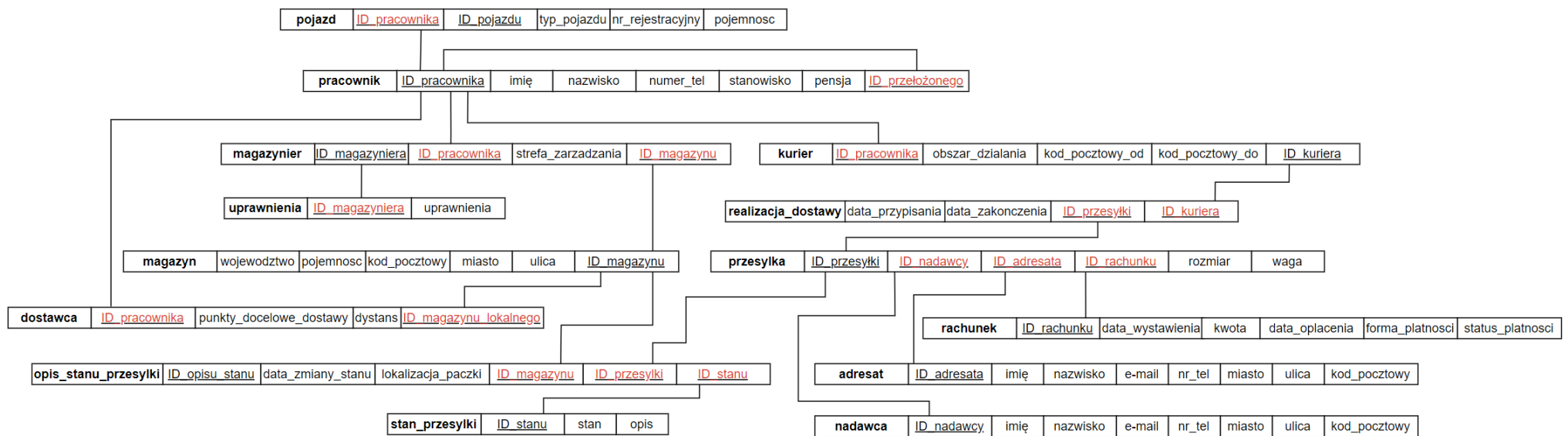
Notacja Barkera



Notacja UML



Model relacyjny



Opis modeli relacyjnych

Wprowadzenie

Przedstawione powyżej modele relacyjne opisują system bazodanowy firmy kurierskiej. W każdym z modeli przedstawione są encje: adresat, dostawca, kurier, magazyn, magazynier, nadawca, opis stanu przesyłki, pojazd, pracownik, przesyłka, rachunek, realizacja dostawy i stan przesyłki.

Założenia

- Pracownik może być kurierem, dostawcą lub magazynierem. Może mieć też inne stanowisko, ale zawsze ma tylko jedno.
- Przesyłka musi mieć adresata, nadawcę, rachunek oraz opis stanu przesyłki.
- Magazynier nie może istnieć bez magazynu. Tak samo dostawca.
- Opis stanu przesyłki musi mieć przesyłkę oraz stan przesyłki. Ewentualnie może mieć również magazyn.
- Rachunek nie może istnieć bez przesyłki.
- Nadawca oraz adresat mogą mieć wiele przesyłek, mogą również nie mieć żadnych.
- Przesyłka, magazyn oraz stan przesyłki mogą mieć wiele opisów stanu przesyłki. Przesyłka musi mieć przynajmniej jeden opis stanu przesyłki, a magazyn i stan przesyłki mogą go nie mieć wcale.
- Magazyn nie musi mieć magazynierów i dostawców, ale może mieć ich wiele.

Założenia poszczególnych encji

Adresat – tabela, która przechowuje informacje o adresatach przesyłek.

ID_adresata – klucz główny

Imię – określa imię adresata

Nazwisko – określa nazwisko adresata

E-mail – określa e-mail adresata

Nr_tel – określa numer telefonu adresata

Miasto – określa miasto, w którym mieszka adresat

Ulica – określa ulicę, na której mieszka adresat

Kod_pocztowy – określa kod pocztowy adresata

Dostawca - tabela, która przechowuje informacje o dostawcach, którzy odpowiadają za transport przesyłek z magazynów do punktu dystrybucyjnego oraz z punktu dystrybucyjnego do poszczególnych magazynów.

Punkty_docelowe_dostawy – określa punkty, do których należy dostarczyć dostawę

Dystans – określa dystans całej trasy

Kurier - tabela, która przechowuje informacje o kurierach, którzy odpowiadają za odbieranie przesyłek od nadawców i dostarczenie ich do odpowiednich magazynów. Odpowiadają również za dostarczenie przesyłki do adresata z danego magazynu.

ID_kuriera – klucz główny

Obszar_dzialania – określa obszar (przeważnie województwo), w którym działa kurier

Kod_pocztowy_od – początek zakresu kodów pocztowych, które obsługuje kurier

Kod_pocztowy_do – koniec zakresu kodów pocztowych, które obsługuje kurier

Magazyn - tabela, która przechowuje informacje o magazynach, w których przechowywane są przesyłki.

ID_magazynu – klucz główny

Miasto – określa miasto, w którym znajduje się magazyn

Ulica – określa ulicę, na której znajduje się magazyn

Pojemnosc – określa pojemność magazynu

Kod_pocztowy – określa kod pocztowy magazynu

Województwo – określa województwo, w którym znajduje się magazyn

Magazynier - tabela, która przechowuje informacje o magazynierach, którzy pracują w magazynach.

ID_magazyniera – klucz główny

Strefa_zarzadzania – określa strefę, którą obsługuje magazynier

Uprawnienia – określa uprawnienia magazyniera (atrybut wielowartościowy)

Nadawca – tabela, która przechowuje informacje o nadawcach przesyłek.

ID_nadawcy – klucz główny

Imię – określa imię nadawcy

Nazwisko – określa nazwisko nadawcy

E-mail – określa e-mail nadawcy

Nr_tel – określa numer telefonu nadawcy

Miasto – określa miasto, w którym mieszka nadawca

Ulica – określa ulicę, na której mieszka nadawca

Kod_pocztowy – określa kod pocztowy nadawcy

Opis stanu przesyłki – tabela, która przechowuje informacje o stanach przesyłki, jej lokalizacji oraz magazynach, w których się znajduje w określonym czasie.

ID_opisu_stanu – klucz główny

Data_zmiany_stanu – określa datę, w której zmienił się stan przesyłki

Lokalizacja_paczki – określa lokalizację przesyłki

Pojazd – tabela, która przechowuje pojazdy pracowników.

ID_pojazdu – klucz główny

Typ_pojazdu – określa typ pojazdu

Nr_rejestacyjny – określa numer rejestracyjny pojazdu

Pojemnosc – określa pojemność pojazdu

Pracownik – tabela, która przechowuje informacje o wszystkich pracownikach firmy.

ID_pracownika – klucz główny

Imię – określa imię pracownika

Nazwisko – określa nazwisko pracownika

Numer_tel – określa numer telefonu pracownika

Stanowisko – określa stanowisko pracownika

Pensja – określa pensję pracownika

Przesyłka – tabela, która przechowuje informacje o przesyłkach.

ID_przesyłki – klucz główny

Waga – określa wagę przesyłki

Rozmiar – określa rozmiar przesyłki

Rachunek – tabela, która przechowuje informacje o rachunku.

ID_rachunku – klucz główny

Data_wystawienia – określa datę wystawienia rachunku

Kwota – określa kwotę rachunku

Forma_platnosci – określa formę płatności za rachunek

Status_platnosci – określa czy rachunek jest opłacony

Data_opłacenia – określa datę opłacenia rachunku

Realizacja dostawy – tabela, która przechowuje informacje o realizacji dostawy przesyłki przez kuriera.

Data_przypisania – określa datę przypisania kuriera do dostarczenia przesyłki

Data_zakończenia – określa datę zakończenia dostawy przesyłki przez kuriera

Stan przesyłki – tabela, która przechowuje informacje o wszystkich dostępnych stanach przesyłki.

ID_stanu – klucz główny

Stan – określa stan przesyłki

Opis – opisuje stan przesyłki

Wszystkie powyższe atrybuty są obowiązkowe oprócz opis w tabeli stan przesyłki, pojemnosc w tabeli pojazd oraz uprawnienia w tabeli magazynier.

Tworzenie tabel i wypełnianie danymi

Poniżej przedstawione są kolejno: stworzenie tabeli, wypełnienie tabeli danymi i wynik w bazie danych.

a) Adresat

Stworzenie tabeli:

```
CREATE TABLE adresat (  
  ID_adresata int(10) NOT NULL AUTO_INCREMENT,  
  imię        varchar(255) NOT NULL,  
  nazwisko    varchar(255) NOT NULL,  
  email       varchar(255) NOT NULL,  
  nr_tel      varchar(20) NOT NULL,  
  miasto      varchar(255) NOT NULL,  
  ulica       varchar(255) NOT NULL,  
  kod_pocztowy varchar(6) NOT NULL,  
  PRIMARY KEY (ID_adresata));
```

Wypełnienie tabeli danymi:

```
INSERT INTO adresat (ID_adresata, imię, nazwisko, email, nr_tel, miasto,  
ulica, kod_pocztowy) VALUES  
  (1, 'Jan', 'Kowalski', 'j.kowalski@gmail.com', '123456789',  
  'Warszawa', 'Ulica 1', '00-001'),  
  (2, 'Anna', 'Nowak', 'a.nowak@gmail.com', '234567890', 'Warszawa',  
  'Ulica 2', '00-015'),  
  (3, 'Marek', 'Zielinski', 'm.zielinski@gmail.com', '345678901',  
  'Kraków', 'Ulica 3', '30-001'),  
  (4, 'Kasia', 'Wojcik', 'k.wojcik@gmail.com', '456789012',  
  'Bydgoszcz', 'Ulica 4', '85-001'),  
  (5, 'Piotr', 'Wisnia', 'p.wisnia@gmail.com', '567890123', 'Kraków',  
  'Ulica 5', '30-020'),  
  (6, 'Ewa', 'Kaczmarek', 'e.kaczmarek@gmail.com', '678901234',  
  'Bydgoszcz', 'Ulica 6', '85-010'),  
  (7, 'Tomasz', 'Dab', 't.dab@gmail.com', '789012345', 'Warszawa',  
  'Ulica 7', '01-234'),  
  (8, 'Agnieszka', 'Aga', 'a.ag@gmail.com', '890123456', 'Olsztyn',  
  'Ulica 8', '10-001'),  
  (9, 'Rafal', 'Nowak', 'r.nowak@gmail.com', '901234567', 'Katowice',  
  'Ulica 9', '40-001'),  
  (10, 'Justyna', 'Kowalczyk', 'j.kowalczyk@gmail.com', '987654321',  
  'Szczecin', 'Ulica 10', '70-001'),  
  (11, 'Jacek', 'Maj', 'j.maj@gmail.com', '876543210', 'Opole', 'Ulica  
11', '45-001'),  
  (12, 'Martyna', 'Mazur', 'm.mazur@gmail.com', '765432109',  
  'Bydgoszcz', 'Ulica 12', '85-020'),  
  (13, 'Jacek', 'Wrobel', 'j.wrobel@gmail.com', '654321098', 'Lublin',  
  'Ulica 13', '20-001'),  
  (14, 'Szymon', 'Pawlak', 's.pawlak@gmail.com', '543210987', 'Kraków',  
  'Ulica 14', '30-030'),  
  (15, 'Natalia', 'Szymczak', 'n.szymczak@gmail.com', '432109876',  
  'Rzeszów', 'Ulica 15', '35-001');
```



Wynik:

ID_adresata	imię	nazwisko	email	nr_tel	miasto	ulica	kod_pocztowy
1	Jan	Kowalski	j.kowalski@gmail.com	123456789	Warszawa	Ulica 1	00-001
2	Anna	Nowak	a.nowak@gmail.com	234567890	Warszawa	Ulica 2	00-015
3	Marek	Zielinski	m.zielinski@gmail.com	345678901	Kraków	Ulica 3	30-001
4	Kasia	Wojcik	k.wojcik@gmail.com	456789012	Bydgoszcz	Ulica 4	85-001
5	Piotr	Wisnia	p.wisnia@gmail.com	567890123	Kraków	Ulica 5	30-020
6	Ewa	Kaczmarek	e.kaczmarek@gmail.com	678901234	Bydgoszcz	Ulica 6	85-010
7	Tomasz	Dab	t.dab@gmail.com	789012345	Warszawa	Ulica 7	01-234
8	Agnieszka	Aga	a.ag@gmail.com	890123456	Olsztyn	Ulica 8	10-001
9	Rafal	Nowak	r.nowak@gmail.com	901234567	Katowice	Ulica 9	40-001
10	Justyna	Kowalczyk	j.kowalczyk@gmail.com	987654321	Szczecin	Ulica 10	70-001
11	Jacek	Maj	j.maj@gmail.com	876543210	Opole	Ulica 11	45-001
12	Martyna	Mazur	m.mazur@gmail.com	765432109	Bydgoszcz	Ulica 12	85-020
13	Jacek	Wrobel	j.wrobel@gmail.com	654321098	Lublin	Ulica 13	20-001
14	Szymon	Pawlak	s.pawlak@gmail.com	543210987	Kraków	Ulica 14	30-030
15	Natalia	Szymczak	n.szymczak@gmail.com	432109876	Rzeszów	Ulica 15	35-001

b) Dostawca

```
CREATE TABLE dostawca(  
  ID_pracownika          int(10) NOT NULL,  
  ID_magazynu_lokalnego   int(10) NOT NULL,  
  punkty_docelowe_dostawy varchar(255) NOT NULL,  
  dystans                 float NOT NULL,  
  FOREIGN KEY (ID_dostawcy) REFERENCES pracownik (ID_pracownika),  
  FOREIGN KEY (ID_magazynu_lokalnego) REFERENCES magazyn (ID_magazynu));
```

```
INSERT INTO dostawca (ID_pracownika, ID_magazynu_lokalnego,  
punkty_docelowe_dostawy, dystans) VALUES  
(41, 1, 'punkt dystrybucyjny', 200),  
(42, 2, 'punkt dystrybucyjny', 321),  
(43, 3, 'punkt dystrybucyjny', 400),  
(44, 4, 'punkt dystrybucyjny', 400),  
(45, 5, 'punkt dystrybucyjny', 320),  
(46, 6, 'punkt dystrybucyjny', 440),  
(47, 7, 'punkt dystrybucyjny', 370),  
(48, 8, 'punkt dystrybucyjny', 380),  
(49, 9, 'punkt dystrybucyjny', 430),  
(50, 10, 'punkt dystrybucyjny', 301),  
(51, 11, 'punkt dystrybucyjny', 229),  
(52, 12, 'punkt dystrybucyjny', 358),  
(53, 13, 'punkt dystrybucyjny', 326),  
(54, 14, 'punkt dystrybucyjny', 400),  
(55, 15, 'punkt dystrybucyjny', 329),  
(56, 15, 'punkt dystrybucyjny', 314),  
(57, 17, 'magazyny - Pomorskie i Kujawsko-Pomorskie', 600),  
(58, 17, 'magazyny - Wielkopolskie i Zachodnio-Pomorskie', 650),  
(59, 17, 'magazyny - Lubuskie i Dolnośląskie', 580),  
(60, 17, 'magazyny - Opolskie i Śląskie', 520),  
(61, 17, 'magazyny - Świętokrzyskie i Lubelskie', 532),  
(62, 17, 'magazyny - Małopolskie i Podkarpackie', 630),  
(63, 17, 'magazyny - Łódzkie i Mazowieckie', 540),  
(64, 17, 'magazyny - Warmińsko-Mazurskie i Podlaskie', 640);
```

#	ID_pracownika 	ID_magazynu_lokalnego 	punkty_docelowe_dostawy	dystans
1	41		1 punkt dystrybucyjny	200
2	42		2 punkt dystrybucyjny	321
3	43		3 punkt dystrybucyjny	400
4	44		4 punkt dystrybucyjny	400
5	45		5 punkt dystrybucyjny	320
6	46		6 punkt dystrybucyjny	440
7	47		7 punkt dystrybucyjny	370
8	48		8 punkt dystrybucyjny	380
9	49		9 punkt dystrybucyjny	430
10	50		10 punkt dystrybucyjny	301
11	51		11 punkt dystrybucyjny	229
12	52		12 punkt dystrybucyjny	358
13	53		13 punkt dystrybucyjny	326
14	54		14 punkt dystrybucyjny	400
15	55		15 punkt dystrybucyjny	329
16	56		15 punkt dystrybucyjny	314
17	57		17 magazyny - Pomorskie i Kujaw...	600
18	58		17 magazyny - Wielkopolskie i Za...	650
19	59		17 magazyny - Lubuskie i Dolnośl...	580
20	60		17 magazyny - Opolskie i Śląskie	520
21	61		17 magazyny - Świętokrzyskie i L...	532
22	62		17 magazyny - Małopolskie i Podk...	630
23	63		17 magazyny - Łódzkie i Mazowie...	540
24	64		17 magazyny - Warmińsko-Mazur...	640

c) Kurier

```
CREATE TABLE kurier (
  ID_pracownika    int(10) NOT NULL,
  obszar_dzialania  varchar(255) NOT NULL,
  kod_pocztowy_od   varchar(6),
  kod_pocztowy_do   varchar(6),
  ID_kuriera        int(10) NOT NULL,
  PRIMARY KEY (ID_kuriera),
  FOREIGN KEY (ID_pracownika) REFERENCES pracownik (ID_pracownika));
```

```
INSERT INTO kurier (ID_pracownika, obszar_dzialania, kod_pocztowy_od,
kod_pocztowy_do, ID_kuriera) VALUES
(4, 'Dolnośląskie', '50-000', '59-999', 1),
(6, 'Kujawsko-Pomorskie', '85-000', '89-999', 2),
(8, 'Lubelskie', '20-000', '24-999', 3),
(10, 'Lubuskie', '65-000', '69-999', 4),
(12, 'Łódzkie', '90-000', '99-999', 5),
(14, 'Małopolskie', '30-000', '34-999', 6),
(16, 'Mazowieckie', '00-000', '09-999', 7),
(22, 'Opolskie', '45-000', '49-999', 8),
(24, 'Podkarpackie', '35-000', '39-999', 9),
(26, 'Podlaskie', '15-000', '19-999', 10),
(28, 'Pomorskie', '80-000', '84-999', 11),
(30, 'Śląskie', '40-000', '44-999', 12),
(32, 'Świętokrzyskie', '25-000', '29-999', 13),
(34, 'Warmińsko-Mazurskie', '10-000', '14-999', 14),
(36, 'Wielkopolskie', '60-000', '64-999', 15),
(38, 'Zachodniopomorskie', '70-000', '79-999', 16),
(40, 'Zastępuje', NULL, NULL, 17);
```

ID_pracownika	obszar_dzialania	kod_pocztowy_od	kod_pocztowy_do	ID_kuriera
4	Dolnośląskie	50-000	59-999	1
6	Kujawsko-Pomorskie	85-000	89-999	2
8	Lubelskie	20-000	24-999	3
10	Lubuskie	65-000	69-999	4
12	Łódzkie	90-000	99-999	5
14	Małopolskie	30-000	34-999	6
16	Mazowieckie	00-000	09-999	7
22	Opolskie	45-000	49-999	8
24	Podkarpackie	35-000	39-999	9
26	Podlaskie	15-000	19-999	10
28	Pomorskie	80-000	84-999	11
30	Śląskie	40-000	44-999	12
32	Świętokrzyskie	25-000	29-999	13
34	Warmińsko-Mazurskie	10-000	14-999	14
36	Wielkopolskie	60-000	64-999	15
38	Zachodniopomorskie	70-000	79-999	16
40	Zastępuje	(NULL)	(NULL)	17

d) Magazyn

```
CREATE TABLE magazyn (
  ID_magazynu int(10) NOT NULL AUTO_INCREMENT,
  miasto      varchar(255) NOT NULL,
  ulica       varchar(255) NOT NULL,
  pojemnosc   int(10),
  kod_pocztowy varchar(255) NOT NULL,
  wojewodztwo varchar(255) NOT NULL,
  PRIMARY KEY (ID_magazynu));
```

```
INSERT INTO magazyn (ID_magazynu, miasto, ulica, pojemnosc, kod_pocztowy,
wojewodztwo) VALUES
(1, 'Warszawa', 'Magazynowa 1', 1000, '00-030', 'Mazowieckie'),
(2, 'Kraków', 'Magazynowa 2', 800, '30-004', 'Małopolskie'),
(3, 'Gdańsk', 'Magazynowa 3', 1200, '80-009', 'Pomorskie'),
(4, 'Wrocław', 'Magazynowa 4', 900, '50-010', 'Dolnośląskie'),
(5, 'Poznań', 'Magazynowa 5', 1100, '60-059', 'Wielkopolskie'),
(6, 'Szczecin', 'Magazynowa 6', 950, '70-002', 'Zachodnio-
Pomorskie'),
(7, 'Lublin', 'Magazynowa 7', 1300, '20-033', 'Lubelskie'),
(8, 'Bydgoszcz', 'Magazynowa 8', 1000, '85-070', 'Kujawsko-
Pomorskie'),
(9, 'Katowice', 'Magazynowa 9', 850, '40-022', 'Śląskie'),
(10, 'Białystok', 'Magazynowa 10', 1200, '15-001', 'Podlaskie'),
(11, 'Olsztyn', 'Magazynowa 11', 950, '10-009', 'Warmińsko-
Mazurskie'),
(12, 'Rzeszów', 'Magazynowa 12', 1100, '35-007', 'Podkarpackie'),
(13, 'Zielona Góra', 'Magazynowa 13', 900, '65-111', 'Lubuskie'),
(14, 'Opole', 'Magazynowa 14', 1150, '45-003', 'Opolskie'),
(15, 'Łódź', 'Magazynowa 15', 1000, '90-005', 'Łódzkie'),
(16, 'Kielce', 'Magazynowa 16', 1050, '25-002', 'Świętokrzyskie'),
(17, 'Piotrków Trybunalski', 'Dystrybucyjna 1', 18000, '97-300',
'Łódzkie');
```

	ID_magazynu	miasto	ulica	pojemnosc	kod_pocztowy	wojewodztwo
1	1	Warszawa	Magazynowa 1	1 000	00-030	Mazowieckie
2	2	Kraków	Magazynowa 2	800	30-004	Małopolskie
3	3	Gdańsk	Magazynowa 3	1 200	80-009	Pomorskie
4	4	Wrocław	Magazynowa 4	900	50-010	Dolnośląskie
5	5	Poznań	Magazynowa 5	1 100	60-059	Wielkopolskie
6	6	Szczecin	Magazynowa 6	950	70-002	Zachodnio-Pomorskie
7	7	Lublin	Magazynowa 7	1 300	20-033	Lubelskie
8	8	Bydgoszcz	Magazynowa 8	1 000	85-070	Kujawsko-Pomorskie
9	9	Katowice	Magazynowa 9	850	40-022	Śląskie
10	10	Białystok	Magazynowa 10	1 200	15-001	Podlaskie
11	11	Olsztyn	Magazynowa 11	950	10-009	Warmińsko-Mazurskie
12	12	Rzeszów	Magazynowa 12	1 100	35-007	Podkarpackie
13	13	Zielona Góra	Magazynowa 13	900	65-111	Lubuskie
14	14	Opole	Magazynowa 14	1 150	45-003	Opolskie
15	15	Łódź	Magazynowa 15	1 000	90-005	Łódzkie
16	16	Kielce	Magazynowa 16	1 050	25-002	Świętokrzyskie
17	17	Piotrków Trybunalski	Dystrybucyjna 1	18 000	97-300	Łódzkie

e) Magazynier

```
CREATE TABLE magazynier (
  ID_pracownika      int(10) NOT NULL,
  strefa_zarzadzania  varchar(255) NOT NULL,
  ID_magazynu         int(10) NOT NULL,
  ID_magazyniera      int(10) NOT NULL,
  PRIMARY KEY (ID_magazyniera),
  FOREIGN KEY (ID_magazynu) REFERENCES magazyn (ID_magazynu),
  FOREIGN KEY (ID_pracownika) REFERENCES pracownik (ID_pracownika));
```

```
INSERT INTO magazynier (ID_pracownika, strefa_zarzadzania, ID_magazynu,
ID_magazyniera) VALUES
  (5, 'strefa_A', 1, 1),
  (7, 'strefa_B', 2, 2),
  (9, 'strefa_C', 3, 3),
  (11, 'strefa_D', 4, 4),
  (13, 'strefa_E', 5, 5),
  (15, 'strefa_F', 6, 6),
  (17, 'strefa_G', 7, 7),
  (23, 'strefa_H', 8, 8),
  (25, 'strefa_I', 9, 9),
  (27, 'strefa_J', 10, 10),
  (29, 'strefa_K', 11, 11),
  (31, 'strefa_L', 12, 12),
  (33, 'strefa_M', 13, 13),
  (35, 'strefa_N', 14, 14),
  (37, 'strefa_O', 17, 15),
  (39, 'strefa_P', 17, 16);
```

ID_pracownika		strefa_zarzadzania	ID_magazynu		ID_magazyniera
5		strefa_A	1		1
7		strefa_B	2		2
9		strefa_C	3		3
11		strefa_D	4		4
13		strefa_E	5		5
15		strefa_F	6		6
17		strefa_G	7		7
23		strefa_H	8		8
25		strefa_I	9		9
27		strefa_J	10		10
29		strefa_K	11		11
31		strefa_L	12		12
33		strefa_M	13		13
35		strefa_N	14		14
37		strefa_O	17		15
39		strefa_P	17		16

f) Nadawca

```
CREATE TABLE nadawca (
  ID_nadawcy    int(10) NOT NULL AUTO_INCREMENT,
  imię          varchar(255) NOT NULL,
  nazwisko      varchar(255) NOT NULL,
  email         varchar(255) NOT NULL,
  nr_tel        varchar(20) NOT NULL,
  miasto        varchar(255) NOT NULL,
  ulica         varchar(255) NOT NULL,
  kod_pocztowy  varchar(6) NOT NULL,
  PRIMARY KEY (ID_nadawcy));
```

```
INSERT INTO nadawca (ID_nadawcy, imię, nazwisko, email, nr_tel, miasto,
ulica, kod_pocztowy) VALUES
(1, 'Adam', 'Adamowski', 'adam.adamowski@gmail.com', '123123123',
'Warszawa', 'Ulica A', '00-001'),
(2, 'Barbara', 'Baran', 'barbara.baran@gmail.com', '234234234',
'Kraków', 'Ulica B', '31-010'),
(3, 'Cezary', 'Cezary', 'cezary.cezary@gmail.com', '345345345',
'Gdańsk', 'Ulica C', '80-342'),
(4, 'Danuta', 'Dąbrowska', 'danuta.dabrowska@gmail.com', '456456456',
'Wrocław', 'Ulica D', '50-003'),
(5, 'Eugeniusz', 'Eugeniusz', 'eugeniusz.eugeniusz@gmail.com',
'567567567', 'Poznań', 'Ulica E', '60-100'),
(6, 'Feliks', 'Felicjański', 'feliks.felicjanski@gmail.com',
'678678678', 'Koszalin', 'Ulica F', '75-021'),
(7, 'Gabriela', 'Gajda', 'gabriela.gajda@gmail.com', '789789789',
'Szczecin', 'Ulica G', '70-981'),
(8, 'Henryk', 'Henrykowski', 'henryk.henrykowski@gmail.com',
'890890890', 'Lublin', 'Ulica H', '20-221'),
(9, 'Iwona', 'Iwanowska', 'iwona.iwanowska@gmail.com', '901901901',
'Bydgoszcz', 'Ulica I', '85-001'),
(10, 'Janusz', 'Jankowski', 'janusz.jankowski@gmail.com',
'987987987', 'Katowice', 'Ulica J', '40-001'),
(11, 'Krzysztof', 'Krysiak', 'krzysztof.krysiak@gmail.com',
'876876876', 'Białystok', 'Ulica K', '15-031'),
(12, 'Lidia', 'Lidzia', 'lidia.lidzia@gmail.com', '765765765',
'Olsztyn', 'Ulica L', '10-001'),
(13, 'Marek', 'Marekowski', 'marek.marekowski@gmail.com',
'654654654', 'Opole', 'Ulica M', '45-001'),
(14, 'Natalia', 'Nowakowska', 'natalia.nowakowska@gmail.com',
'543543543', 'Rzeszów', 'Ulica N', '35-001'),
(15, 'Oliwia', 'Oliwińska', 'oliwia.oliwinska@gmail.com',
'432432432', 'Toruń', 'Ulica O', '87-001');
```

ID_nadawcy	imię	nazwisko	email	nr_tel	miasto	ulica	kod_pocztowy
1	Adam	Adamowski	adam.adamowski@gmail.com	123123123	Warszawa	Ulica A	00-001
2	Barbara	Baran	barbara.baran@gmail.com	234234234	Kraków	Ulica B	31-010
3	Cezary	Cezary	cezary.cezary@gmail.com	345345345	Gdańsk	Ulica C	80-342
4	Danuta	Dąbrowska	danuta.dabrowska@gmail.com	456456456	Wrocław	Ulica D	50-003
5	Eugeniusz	Eugeniusz	eugeniusz.eugeniusz@gmail.c...	567567567	Poznań	Ulica E	60-100
6	Feliks	Felicjański	feliks.felicjanski@gmail.com	678678678	Koszalin	Ulica F	75-021
7	Gabriela	Gajda	gabriela.gajda@gmail.com	789789789	Szczecin	Ulica G	70-981
8	Henryk	Henrykowski	henryk.henrykowski@gmail.com	890890890	Lublin	Ulica H	20-221
9	Iwona	Iwanowska	iwona.iwanowska@gmail.com	901901901	Bydgoszcz	Ulica I	85-001
10	Janusz	Jankowski	janusz.jankowski@gmail.com	987987987	Katowice	Ulica J	40-001
11	Krzysztof	Krysiak	krzysztof.krysiak@gmail.com	876876876	Białystok	Ulica K	15-031
12	Lidia	Lidzia	lidia.lidzia@gmail.com	765765765	Olsztyn	Ulica L	10-001
13	Marek	Marekowski	marek.marekowski@gmail.com	654654654	Opole	Ulica M	45-001
14	Natalia	Nowakowska	natalia.nowakowska@gmail.com	543543543	Rzeszów	Ulica N	35-001
15	Oliwia	Oliwińska	oliwia.oliwinska@gmail.com	432432432	Toruń	Ulica O	87-001

g) Opis stanu przesyłki

```
CREATE TABLE opis_stanu_przesylki (
  ID_opisu_stanu      int(10) NOT NULL AUTO_INCREMENT,
  data_zmiany_stanu   datetime NOT NULL,
  lokalizacja_paczki  varchar(255),
  ID_przesyłki        int(10) NOT NULL,
  ID_stanu             int(10) NOT NULL,
  ID_magazynu         int(10),
  PRIMARY KEY (ID_opisu_stanu),
  FOREIGN KEY (ID_przesyłki) REFERENCES przesyłka (ID_przesyłki),
  FOREIGN KEY (ID_stanu) REFERENCES stan_przesyłki (ID_stanu),
  FOREIGN KEY (ID_magazynu) REFERENCES magazyn (ID_magazynu));
```

```
INSERT INTO opis_stanu_przesylki (ID_opisu_stanu, data_zmiany_stanu,
lokalizacja_paczki, ID_przesyłki, ID_stanu, ID_magazynu) VALUES
(1, '2024-11-03 12:30:00', 'u nadawcy', 1, 1, NULL),
(2, '2024-11-03 09:00:00', 'u nadawcy', 2, 1, NULL),
(3, '2024-11-03 20:00:00', 'u nadawcy', 3, 1, NULL),
(4, '2024-11-03 21:15:00', 'u nadawcy', 4, 1, NULL),
(5, '2024-11-03 21:00:00', 'u nadawcy', 5, 1, NULL),
(6, '2024-11-03 23:00:00', 'u nadawcy', 6, 1, NULL),
(7, '2024-11-03 20:00:00', 'u nadawcy', 7, 1, NULL),
(8, '2024-11-03 19:00:00', 'u nadawcy', 8, 1, NULL),
(9, '2024-11-03 18:00:00', 'u nadawcy', 9, 1, NULL),
(10, '2024-11-03 20:00:00', 'u nadawcy', 10, 1, NULL),
(11, '2024-11-04 20:20:00', 'u nadawcy', 11, 1, NULL),
(12, '2024-11-04 09:00:00', 'u nadawcy', 12, 1, NULL),
(13, '2024-11-04 15:00:00', 'u nadawcy', 13, 1, NULL),
(14, '2024-11-04 16:00:00', 'u nadawcy', 14, 1, NULL),
(15, '2024-11-04 17:00:00', 'u nadawcy', 15, 1, NULL),
(16, '2024-11-03 12:59:30', 'u nadawcy', 1, 2, NULL),
(17, '2024-11-04 10:00:09', 'u nadawcy', 2, 2, NULL),
(18, '2024-11-03 20:30:00', 'u nadawcy', 3, 2, NULL),
(19, '2024-11-03 21:21:21', 'u nadawcy', 4, 2, NULL),
(20, '2024-11-04 10:10:10', 'u nadawcy', 5, 2, NULL),
(21, '2024-11-03 23:23:23', 'u nadawcy', 6, 2, NULL),
(22, '2024-11-03 20:20:20', 'u nadawcy', 7, 2, NULL),
(23, '2024-11-03 19:19:19', 'u nadawcy', 8, 2, NULL),
(24, '2024-11-03 18:18:18', 'u nadawcy', 9, 2, NULL),
(25, '2024-11-04 10:15:10', 'u nadawcy', 10, 2, NULL),
(26, '2024-11-04 11:59:30', 'w drodze', 1, 3, NULL),
(27, '2024-11-05 11:00:09', 'w drodze', 2, 3, NULL),
(28, '2024-11-04 11:30:00', 'w drodze', 3, 3, NULL),
(29, '2024-11-04 10:21:21', 'w drodze', 4, 3, NULL),
(30, '2024-11-05 11:10:10', 'w drodze', 5, 3, NULL),
(31, '2024-11-04 11:23:23', 'w drodze', 6, 3, NULL),
(32, '2024-11-04 10:20:20', 'w drodze', 7, 3, NULL),
(33, '2024-11-04 10:19:19', 'w drodze', 8, 3, NULL),
(34, '2024-11-04 11:18:18', 'w drodze', 9, 3, NULL),
(35, '2024-11-04 10:15:10', 'w drodze', 10, 3, NULL),
(36, '2024-11-05 15:20:29', 'w magazynie', 2, 5, 1),
(37, '2024-11-04 14:58:54', 'w magazynie', 3, 5, 1),
(38, '2024-11-04 15:19:54', 'w magazynie', 1, 5, 2),
(39, '2024-11-04 15:35:32', 'w magazynie', 4, 5, 4),
(40, '2024-11-05 13:10:02', 'w magazynie', 5, 5, 5),
(41, '2024-11-04 15:44:21', 'w magazynie', 6, 5, 6),
(42, '2024-11-04 14:12:27', 'w magazynie', 7, 5, 6),
(43, '2024-11-04 14:55:35', 'w magazynie', 8, 5, 7),
(44, '2024-11-04 13:10:04', 'w magazynie', 9, 5, 8),
(45, '2024-11-04 13:39:13', 'w magazynie', 10, 5, 9),
```

```

(46, '2024-11-04 17:59:30', 'w punkcie dystrybucyjnym', 1, 8, 17),
(47, '2024-11-04 19:00:00', 'w punkcie dystrybucyjnym', 3, 8, 17),
(48, '2024-11-04 18:12:30', 'w punkcie dystrybucyjnym', 4, 8, 17),
(49, '2024-11-05 19:10:10', 'w punkcie dystrybucyjnym', 5, 8, 17),
(50, '2024-11-04 18:23:23', 'w punkcie dystrybucyjnym', 6, 8, 17),
(51, '2024-11-04 18:40:00', 'w punkcie dystrybucyjnym', 7, 8, 17),
(52, '2024-11-04 17:59:00', 'w punkcie dystrybucyjnym', 8, 8, 17),
(53, '2024-11-04 19:18:30', 'w punkcie dystrybucyjnym', 9, 8, 17),
(54, '2024-11-04 19:20:20', 'w punkcie dystrybucyjnym', 10, 8, 17),
(55, '2024-11-04 20:59:30', 'w magazynie docelowym', 1, 14, 1),
(56, '2024-11-04 20:40:00', 'w magazynie docelowym', 7, 14, 1),
(57, '2024-11-04 21:30:00', 'w magazynie docelowym', 3, 14, 2),
(58, '2024-11-05 21:10:10', 'w magazynie docelowym', 5, 14, 2),
(59, '2024-11-04 20:12:30', 'w magazynie docelowym', 4, 14, 8),
(60, '2024-11-04 20:23:23', 'w magazynie docelowym', 6, 14, 8),
(61, '2024-11-04 21:18:30', 'w magazynie docelowym', 9, 14, 9),
(62, '2024-11-04 19:59:00', 'w magazynie docelowym', 8, 14, 11),
(63, '2024-11-05 08:00:00', 'w drodze do adresata', 6, 12, NULL),
(64, '2024-11-05 10:00:00', 'w drodze do adresata', 4, 12, NULL),
(65, '2024-11-06 08:00:00', 'w drodze do adresata', 5, 12, NULL),
(66, '2024-11-05 10:00:00', 'w drodze do adresata', 3, 12, NULL),
(67, '2024-11-05 08:00:00', 'w drodze do adresata', 7, 12, NULL),
(68, '2024-11-06 09:00:00', 'w drodze do adresata', 2, 12, NULL),
(69, '2024-11-05 08:03:00', 'w drodze do adresata', 1, 12, NULL),
(70, '2024-11-05 09:30:00', 'w drodze do adresata', 9, 12, NULL),
(71, '2024-11-05 09:00:00', 'w drodze do adresata', 8, 12, NULL),
(72, '2024-11-05 09:17:40', 'dostarczona', 6, 15, NULL),
(73, '2024-11-05 11:49:30', 'dostarczona', 4, 15, NULL),
(74, '2024-11-06 10:01:55', 'dostarczona', 5, 15, NULL),
(75, '2024-11-05 12:33:27', 'dostarczona', 3, 15, NULL),
(76, '2024-11-05 10:02:22', 'dostarczona', 7, 15, NULL),
(77, '2024-11-06 11:19:58', 'dostarczona', 2, 15, NULL),
(78, '2024-11-05 09:52:29', 'dostarczona', 1, 15, NULL),
(79, '2024-11-05 11:26:49', 'dostarczona', 9, 15, NULL),
(80, '2024-11-05 11:04:14', 'dostarczona', 8, 15, NULL),
(81, '2024-11-05 17:39:56', 'w punkcie dystrybucyjnym', 2, 8, 17),
(82, '2024-11-05 21:12:37', 'w magazynie docelowym', 2, 14, 1);

```

	ID_opisu_stanu 📄	data_zmiany_stanu 📅	lokalizacja_paczki	ID_przesyłki 📄	ID_stanu 📄	ID_magazynu 📄
1	1	2024-11-03 12:30:00	u nadawcy	1	1	(NULL)
2	16	2024-11-03 12:59:30	u nadawcy	1	2	(NULL)
3	26	2024-11-04 11:59:30	w drodze	1	3	(NULL)
4	38	2024-11-04 15:19:54	w magazynie	1	5	2
5	46	2024-11-04 17:59:30	w punkcie dystrybucyjnym	1	8	17
6	55	2024-11-04 20:59:30	w magazynie docelowym	1	14	1
7	69	2024-11-05 08:03:00	w drodze do adresata	1	12	(NULL)
8	78	2024-11-05 09:52:29	dostarczona	1	15	(NULL)
9	2	2024-11-03 09:00:00	u nadawcy	2	1	(NULL)
10	17	2024-11-04 10:00:09	u nadawcy	2	2	(NULL)
11	27	2024-11-05 11:00:09	w drodze	2	3	(NULL)
12	36	2024-11-05 15:20:29	w magazynie	2	5	1
13	81	2024-11-05 17:39:56	w punkcie dystrybucyjnym	2	8	17
14	82	2024-11-05 21:12:37	w magazynie docelowym	2	14	1
15	68	2024-11-06 09:00:00	w drodze do adresata	2	12	(NULL)
16	77	2024-11-06 11:19:58	dostarczona	2	15	(NULL)
17	3	2024-11-03 20:00:00	u nadawcy	3	1	(NULL)
18	18	2024-11-03 20:30:00	u nadawcy	3	2	(NULL)
19	28	2024-11-04 11:30:00	w drodze	3	3	(NULL)
20	37	2024-11-04 14:58:54	w magazynie	3	5	1
21	47	2024-11-04 19:00:00	w punkcie dystrybucyjnym	3	8	17
22	57	2024-11-04 21:30:00	w magazynie docelowym	3	14	2
23	66	2024-11-05 10:00:00	w drodze do adresata	3	12	(NULL)
24	75	2024-11-05 12:33:27	dostarczona	3	15	(NULL)

	ID_opisu_stanu	data_zmiany_stanu	lokalizacja_paczki	ID_przesyłki	ID_stanu	ID_magazynu
25	4	2024-11-03 21:15:00	u nadawcy	4	1	(NULL)
26	19	2024-11-03 21:21:21	u nadawcy	4	2	(NULL)
27	29	2024-11-04 10:21:21	w drodze	4	3	(NULL)
28	39	2024-11-04 15:35:32	w magazynie	4	5	4
29	48	2024-11-04 18:12:30	w punkcie dystrybucyjnym	4	8	17
30	59	2024-11-04 20:12:30	w magazynie docelowym	4	14	8
31	64	2024-11-05 10:00:00	w drodze do adresata	4	12	(NULL)
32	73	2024-11-05 11:49:30	dostarczona	4	15	(NULL)
33	5	2024-11-03 21:00:00	u nadawcy	5	1	(NULL)
34	20	2024-11-04 10:10:10	u nadawcy	5	2	(NULL)
35	30	2024-11-05 11:10:10	w drodze	5	3	(NULL)
36	40	2024-11-05 13:10:02	w magazynie	5	5	5
37	49	2024-11-05 19:10:10	w punkcie dystrybucyjnym	5	8	17
38	58	2024-11-05 21:10:10	w magazynie docelowym	5	14	2
39	65	2024-11-06 08:00:00	w drodze do adresata	5	12	(NULL)
40	74	2024-11-06 10:01:55	dostarczona	5	15	(NULL)
41	6	2024-11-03 23:00:00	u nadawcy	6	1	(NULL)
42	21	2024-11-03 23:23:23	u nadawcy	6	2	(NULL)
43	31	2024-11-04 11:23:23	w drodze	6	3	(NULL)
44	41	2024-11-04 15:44:21	w magazynie	6	5	6
45	50	2024-11-04 18:23:23	w punkcie dystrybucyjnym	6	8	17
46	60	2024-11-04 20:23:23	w magazynie docelowym	6	14	8
47	63	2024-11-05 08:00:00	w drodze do adresata	6	12	(NULL)
48	72	2024-11-05 09:17:40	dostarczona	6	15	(NULL)
49	7	2024-11-03 20:00:00	u nadawcy	7	1	(NULL)
50	22	2024-11-03 20:20:20	u nadawcy	7	2	(NULL)
51	32	2024-11-04 10:20:20	w drodze	7	3	(NULL)
52	42	2024-11-04 14:12:27	w magazynie	7	5	6
53	51	2024-11-04 18:40:00	w punkcie dystrybucyjnym	7	8	17
54	56	2024-11-04 20:40:00	w magazynie docelowym	7	14	1
55	67	2024-11-05 08:00:00	w drodze do adresata	7	12	(NULL)
56	76	2024-11-05 10:02:22	dostarczona	7	15	(NULL)
57	8	2024-11-03 19:00:00	u nadawcy	8	1	(NULL)
58	23	2024-11-03 19:19:19	u nadawcy	8	2	(NULL)
59	33	2024-11-04 10:19:19	w drodze	8	3	(NULL)
60	43	2024-11-04 14:55:35	w magazynie	8	5	7
61	52	2024-11-04 17:59:00	w punkcie dystrybucyjnym	8	8	17
62	62	2024-11-04 19:59:00	w magazynie docelowym	8	14	11
63	71	2024-11-05 09:00:00	w drodze do adresata	8	12	(NULL)
64	80	2024-11-05 11:04:14	dostarczona	8	15	(NULL)
65	9	2024-11-03 18:00:00	u nadawcy	9	1	(NULL)
66	24	2024-11-03 18:18:18	u nadawcy	9	2	(NULL)
67	34	2024-11-04 11:18:18	w drodze	9	3	(NULL)
68	44	2024-11-04 13:10:04	w magazynie	9	5	8
69	53	2024-11-04 19:18:30	w punkcie dystrybucyjnym	9	8	17
70	61	2024-11-04 21:18:30	w magazynie docelowym	9	14	9
71	70	2024-11-05 09:30:00	w drodze do adresata	9	12	(NULL)
72	79	2024-11-05 11:26:49	dostarczona	9	15	(NULL)
73	10	2024-11-03 20:00:00	u nadawcy	10	1	(NULL)
74	35	2024-11-04 10:15:10	w drodze	10	3	(NULL)
75	25	2024-11-04 10:15:10	u nadawcy	10	2	(NULL)
76	45	2024-11-04 13:39:13	w magazynie	10	5	9
77	54	2024-11-04 19:20:20	w punkcie dystrybucyjnym	10	8	17
78	11	2024-11-04 20:20:00	u nadawcy	11	1	(NULL)
79	12	2024-11-04 09:00:00	u nadawcy	12	1	(NULL)
80	13	2024-11-04 15:00:00	u nadawcy	13	1	(NULL)
81	14	2024-11-04 16:00:00	u nadawcy	14	1	(NULL)
82	15	2024-11-04 17:00:00	u nadawcy	15	1	(NULL)

h) Pojazd

```
CREATE TABLE pojazd (  
  ID_pojazdu      int(10) NOT NULL AUTO_INCREMENT,  
  typ_pojazdu     varchar(255) NOT NULL,  
  nr_rejestracyjny varchar(255) NOT NULL UNIQUE,  
  pojemnosc       int(10),  
  ID_pracownika   int(10),  
  PRIMARY KEY (ID_pojazdu),  
  FOREIGN KEY (ID_pracownika) REFERENCES pracownik (ID_pracownika));
```

```
INSERT INTO pojazd (ID_pojazdu, typ_pojazdu, nr_rejestracyjny, pojemnosc,  
ID_pracownika) VALUES  
(1, 'Bus', 'ZXC12345', 2500, 4),  
(2, 'Bus', 'RTY45678', 3000, 6),  
(3, 'Samochód dostawczy', 'YUI78901', 2000, 8),  
(4, 'Bus', 'QWE12345', 2800, 10),  
(5, 'Bus', 'ASD45678', 2500, 12),  
(6, 'Samochód dostawczy', 'FGH78901', 2200, 14),  
(7, 'Bus', 'HJK12345', 3000, 16),  
(8, 'Van', 'BNJ45678', 2700, 22),  
(9, 'Bus', 'MNB78901', 2500, 24),  
(10, 'Bus', 'VFR12345', 2800, 26),  
(11, 'Samochód dostawczy', 'TGW45678', 2200, 28),  
(12, 'Bus', 'ZHN78901', 3000, 30),  
(13, 'Bus', 'PLM12345', 2500, 32),  
(14, 'Bus', 'QAZ45678', 2500, 34),  
(15, 'Samochód dostawczy', 'WSX78901', 2300, 36),  
(16, 'Bus', 'EDC12345', 2500, 38),  
(17, 'Bus', 'RFV45678', 2800, 40),  
(18, 'Samochód dostawczy', 'TGB78901', 2200, 16),  
(19, 'Bus', 'YHN12345', 3000, 41),  
(20, 'Van', 'UJM45678', 2600, 42),  
(21, 'Bus', 'IKL78901', 2500, 43),  
(22, 'Bus', 'OPZ12345', 2800, 44),  
(23, 'Samochód dostawczy', 'XYZ45678', 2000, 45),  
(24, 'Bus', 'NHG45678', 2500, 46),  
(25, 'Bus', 'BNM78901', 2500, 47),  
(26, 'Bus', 'LKI12345', 2700, 48),  
(27, 'Van', 'POI78901', 2400, 49),  
(28, 'Bus', 'UYT12345', 2500, 50),  
(29, 'Bus', 'TRE45678', 2800, 51),  
(30, 'Van', 'QAZ78901', 2600, 52),  
(31, 'Van', 'ZXA44444', 2500, 53),  
(32, 'Bus', 'RTY55555', 2700, 54),  
(33, 'Samochód dostawczy', 'YUI66666', 2200, 55),  
(34, 'Bus', 'QWE77777', 2800, 56),  
(35, 'Bus', 'LIM12342', 2800, 57),  
(36, 'Samochód dostawczy', 'SAM98765', 3000, 58),  
(37, 'Van', 'TGV98765', 2400, 59),  
(38, 'Bus', 'BNQ87654', 2500, 60),  
(39, 'Samochód dostawczy', 'PLM65432', 2000, 61),  
(40, 'Bus', 'ASX54321', 2800, 62),  
(41, 'Van', 'UJM43210', 2600, 63),  
(42, 'Bus', 'KLM32109', 2700, 64),  
(43, 'Samochód dostawczy', 'BVC21098', 3000, NULL);
```

	ID_pojazdu	typ_pojazdu	nr_rejestracyjny	pojemnosc	ID_pracownika
1	1	Bus	ZXC12345	2 500	4
2	2	Bus	RTY45678	3 000	6
3	3	Samochód dostawczy	YUI78901	2 000	8
4	4	Bus	QWE12345	2 800	10
5	5	Bus	ASD45678	2 500	12
6	6	Samochód dostawczy	FGH78901	2 200	14
7	7	Bus	HJK12345	3 000	16
8	8	Van	BNJ45678	2 700	22
9	9	Bus	MNB78901	2 500	24
10	10	Bus	VFR12345	2 800	26
11	11	Samochód dostawczy	TGW45678	2 200	28
12	12	Bus	ZHN78901	3 000	30
13	13	Bus	PLM12345	2 500	32
14	14	Bus	QAZ45678	2 500	34
15	15	Samochód dostawczy	WSX78901	2 300	36
16	16	Bus	EDC12345	2 500	38
17	17	Bus	RFV45678	2 800	40
18	18	Samochód dostawczy	TGB78901	2 200	16
19	19	Bus	YHN12345	3 000	41
20	20	Van	UJM45678	2 600	42
21	21	Bus	IKL78901	2 500	43
22	22	Bus	OPZ12345	2 800	44
23	23	Samochód dostawczy	XYZ45678	2 000	45
24	24	Bus	NHG45678	2 500	46
25	25	Bus	BNM78901	2 500	47
26	26	Bus	LKI12345	2 700	48
27	27	Van	POI78901	2 400	49
28	28	Bus	UYT12345	2 500	50
29	29	Bus	TRE45678	2 800	51
30	30	Van	QAZ78901	2 600	52
31	31	Van	ZXA44444	2 500	53
32	32	Bus	RTY55555	2 700	54
33	33	Samochód dostawczy	YUI66666	2 200	55
34	34	Bus	QWE77777	2 800	56
35	35	Bus	LIM12342	2 800	57
36	36	Samochód dostawczy	SAM98765	3 000	58
37	37	Van	TGV98765	2 400	59
38	38	Bus	BNQ87654	2 500	60
39	39	Samochód dostawczy	PLM65432	2 000	61
40	40	Bus	ASX54321	2 800	62
41	41	Van	UJM43210	2 600	63
42	42	Bus	KLM32109	2 700	64
43	43	Samochód dostawczy	BVC21098	3 000	(NULL)

i) Pracownik



```
CREATE TABLE pracownik (
  ID_pracownika    int(10) NOT NULL AUTO_INCREMENT,
  imię              varchar(255) NOT NULL,
  nazwisko          varchar(255) NOT NULL,
  numer_tel         varchar(20) NOT NULL,
  stanowisko        varchar(255) NOT NULL,
  pensja            float NOT NULL,
  ID_przełożonego  int(10),
  PRIMARY KEY (ID_pracownika),
  FOREIGN KEY (ID_przełożonego) REFERENCES pracownik (ID_pracownika));
```



```

INSERT INTO pracownik (ID_pracownika, imię, nazwisko, numer_tel, stanowisko,
pensja, ID_przełożonego) VALUES
(1, 'Adam', 'Nowak', '123456789', 'Dyrektor', 12000, NULL),
(2, 'Bartek', 'Zielinski', '987654321', 'Kierownik', 5200, 1),
(3, 'Cezary', 'Adamski', '564738912', 'Kierownik', 5200, 1),
(4, 'Dawid', 'Wojcik', '987456321', 'Kurier', 3000, 2),
(5, 'Ewa', 'Sikora', '948456789', 'Magazynier', 3200, 3),
(6, 'Filip', 'Kowalczyk', '786431987', 'Kurier', 3100, 2),
(7, 'Grzegorz', 'Zawadzki', '984761234', 'Magazynier', 3200, 3),
(8, 'Hanna', 'Mazur', '987654321', 'Kurier', 3100, 2),
(9, 'Iwona', 'Czajka', '987456321', 'Magazynier', 3200, 3),
(10, 'Jan', 'Olszewski', '987123654', 'Kurier', 3000, 2),
(11, 'Kamil', 'Duda', '987321456', 'Magazynier', 3300, 3),
(12, 'Lukasz', 'Pacek', '456789654', 'Kurier', 3000, 2),
(13, 'Marta', 'Szymczak', '654789312', 'Magazynier', 3150, 3),
(14, 'Natalia', 'Szymczak', '321987654', 'Kurier', 3000, 2),
(15, 'Oliwia', 'Czerwona', '789123456', 'Magazynier', 3150, 3),
(16, 'Piotr', 'Bielecki', '321654987', 'Kurier', 3050, 2),
(17, 'Rafal', 'Szczepaniak', '789321456', 'Magazynier', 3050, 3),
(18, 'Szymon', 'Gorski', '564123987', 'Infolinia', 3300, 1),
(19, 'Tomasz', 'Chmielewski', '987654123', 'Pomoc dla klientów', 3400, 1),
(20, 'Urszula', 'Jakubowska', '321789654', 'Infolinia', 3450, 1),
(21, 'Wojciech', 'Lis', '567894321', 'Pomoc dla klientów', 3500, 1),
(22, 'Zofia', 'Krol', '123987654', 'Kurier', 3200, 2),
(23, 'Pawel', 'Sawicki', '654789123', 'Magazynier', 3100, 3),
(24, 'Jakub', 'Jankowski', '987456321', 'Kurier', 3000, 2),
(25, 'Karolina', 'Lesniak', '321654987', 'Magazynier', 3200, 3),
(26, 'Barbara', 'Turek', '654321987', 'Kurier', 3100, 2),
(27, 'Krystian', 'Borowski', '789123987', 'Magazynier', 3000, 3),
(28, 'Magdalena', 'Wilk', '321987654', 'Kurier', 3000, 2),
(29, 'Emilia', 'Blaszczyk', '789654123', 'Magazynier', 3050, 3),
(30, 'Kacper', 'Pawlak', '789654123', 'Kurier', 3100, 2),
(31, 'Olga', 'Kowal', '654789654', 'Magazynier', 3050, 3),
(32, 'Oskar', 'Jablonski', '321456789', 'Kurier', 3100, 2),
(33, 'Patryk', 'Gajda', '987321654', 'Magazynier', 3200, 3),
(34, 'Robert', 'Broza', '987654321', 'Kurier', 3000, 2),
(35, 'Sylwia', 'Kaminska', '321456987', 'Magazynier', 3000, 3),
(36, 'Tadeusz', 'Stankiewicz', '123456789', 'Kurier', 3150, 2),
(37, 'Wiktoria', 'Mocek', '321456789', 'Magazynier', 3150, 3),
(38, 'Zbigniew', 'Polak', '321654789', 'Kurier', 3200, 2),
(39, 'Janusz', 'Zagorski', '987654321', 'Magazynier', 3200, 3),
(40, 'Maria', 'Sowa', '987654654', 'Kurier', 3000, 2),
(41, 'Alicja', 'Kowalska', '321789987', 'Dostawca', 3100, 2),
(42, 'Bartosz', 'Malinowski', '789456123', 'Dostawca', 3000, 2),
(43, 'Cezary', 'Nowicki', '654123789', 'Dostawca', 3150, 2),
(44, 'Daria', 'Gorska', '789321654', 'Dostawca', 3100, 2),
(45, 'Edward', 'Jurek', '321789654', 'Dostawca', 3000, 2),
(46, 'Franciszek', 'Rataj', '456987321', 'Dostawca', 3100, 2),
(47, 'Gabriela', 'Król', '654321123', 'Dostawca', 3050, 2),
(48, 'Hubert', 'Zakrzewski', '789654789', 'Dostawca', 3150, 2),
(49, 'Izabela', 'Baran', '123456987', 'Dostawca', 3100, 2),
(50, 'Jacek', 'Dąbrowski', '456123789', 'Dostawca', 3000, 2),
(51, 'Katarzyna', 'Polanska', '789123654', 'Dostawca', 3150, 2),
(52, 'Leon', 'Ostrowski', '321654321', 'Dostawca', 3050, 2),
(53, 'Marcelina', 'Zięba', '456789123', 'Dostawca', 3200, 2),
(54, 'Norbert', 'Wieczorek', '654321456', 'Dostawca', 3000, 2),
(55, 'Olaf', 'Wróbel', '987321123', 'Dostawca', 3100, 2),
(56, 'Paulina', 'Kucharska', '321789123', 'Dostawca', 3000, 2),
(57, 'Ryszard', 'Bartosz', '654123456', 'Dostawca', 3150, 2),
(58, 'Krzysztof', 'Nowak', '123789456', 'Dostawca', 3100, 2),
(59, 'Lidia', 'Zajac', '789654987', 'Dostawca', 3050, 2),
(60, 'Michał', 'Gajewski', '321654789', 'Dostawca', 3150, 2),
(61, 'Nina', 'Dębowska', '456789987', 'Dostawca', 3000, 2),
(62, 'Oliwier', 'Pietrzak', '654987123', 'Dostawca', 3200, 2),
(63, 'Piotr', 'Mazowiecki', '789123987', 'Dostawca', 3100, 2),
(64, 'Sara', 'Lisowska', '123654987', 'Dostawca', 3050, 2);

```

	ID_pracownika		imię	nazwisko	numer_tel	stanowisko	pensja	ID_przełożonego	
1	1		Adam	Nowak	123456789	Dyrektor	12 000	(NULL)	
2	2		Bartek	Zielinski	987654321	Kierownik	5 200	1	
3	3		Cezary	Adamski	564738912	Kierownik	5 200	1	
4	4		Dawid	Wojcik	987456321	Kurier	3 000	2	
5	5		Ewa	Sikora	948456789	Magazynier	3 200	3	
6	6		Filip	Kowalczyk	786431987	Kurier	3 100	2	
7	7		Grzegorz	Zawadzki	984761234	Magazynier	3 200	3	
8	8		Hanna	Mazur	987654321	Kurier	3 100	2	
9	9		Iwona	Czajka	987456321	Magazynier	3 200	3	
10	10		Jan	Olszewski	987123654	Kurier	3 000	2	
11	11		Kamil	Duda	987321456	Magazynier	3 300	3	
12	12		Lukasz	Pacek	456789654	Kurier	3 000	2	
13	13		Marta	Szymczak	654789312	Magazynier	3 150	3	
14	14		Natalia	Szymczak	321987654	Kurier	3 000	2	
15	15		Oliwia	Czerwonka	789123456	Magazynier	3 150	3	
16	16		Piotr	Bielecki	321654987	Kurier	3 050	2	
17	17		Rafal	Szczepaniak	789321456	Magazynier	3 050	3	
18	18		Szymon	Gorski	564123987	Infolinia	3 300	1	
19	19		Tomasz	Chmielewski	987654123	Pomoc dla klientów	3 400	1	
20	20		Urszula	Jakubowska	321789654	Infolinia	3 450	1	
21	21		Wojciech	Lis	567894321	Pomoc dla klientów	3 500	1	
22	22		Zofia	Krol	123987654	Kurier	3 200	2	
23	23		Pawel	Sawicki	654789123	Magazynier	3 100	3	
24	24		Jakub	Jankowski	987456321	Kurier	3 000	2	
25	25		Karolina	Lesniak	321654987	Magazynier	3 200	3	
26	26		Barbara	Turek	654321987	Kurier	3 100	2	
27	27		Krystian	Borowski	789123987	Magazynier	3 000	3	
28	28		Magdalena	Wilk	321987654	Kurier	3 000	2	
29	29		Emilia	Blaszczyk	789654123	Magazynier	3 050	3	
30	30		Kacper	Pawlak	789654123	Kurier	3 100	2	
31	31		Olga	Kowal	654789654	Magazynier	3 050	3	
32	32		Oskar	Jablonski	321456789	Kurier	3 100	2	
33	33		Patryk	Gajda	987321654	Magazynier	3 200	3	
34	34		Robert	Broza	987654321	Kurier	3 000	2	
35	35		Sylvia	Kaminska	321456987	Magazynier	3 000	3	
36	36		Tadeusz	Stankiewicz	123456789	Kurier	3 150	2	
37	37		Wiktoria	Mocek	321456789	Magazynier	3 150	3	
38	38		Zbigniew	Polak	321654789	Kurier	3 200	2	
39	39		Janusz	Zagorski	987654321	Magazynier	3 200	3	
40	40		Maria	Sowa	987654654	Kurier	3 000	2	
41	41		Alicja	Kowalska	321789987	Dostawca	3 100	2	
42	42		Bartosz	Malinowski	789456123	Dostawca	3 000	2	
43	43		Cezary	Nowicki	654123789	Dostawca	3 150	2	
44	44		Daria	Gorska	789321654	Dostawca	3 100	2	
45	45		Edward	Jurek	321789654	Dostawca	3 000	2	
46	46		Franciszek	Rataj	456987321	Dostawca	3 100	2	
47	47		Gabriela	Król	654321123	Dostawca	3 050	2	
48	48		Hubert	Zakrzewski	789654789	Dostawca	3 150	2	
49	49		Izabela	Baran	123456987	Dostawca	3 100	2	
50	50		Jacek	Dąbrowski	456123789	Dostawca	3 000	2	
51	51		Katarzyna	Polanska	789123654	Dostawca	3 150	2	
52	52		Leon	Ostrowski	321654321	Dostawca	3 050	2	
53	53		Marcelina	Zięba	456789123	Dostawca	3 200	2	
54	54		Norbert	Wieczorek	654321456	Dostawca	3 000	2	
55	55		Olaf	Wróbel	987321123	Dostawca	3 100	2	
56	56		Paulina	Kucharska	321789123	Dostawca	3 000	2	
57	57		Ryszard	Bartosz	654123456	Dostawca	3 150	2	
58	58		Krzysztof	Nowak	123789456	Dostawca	3 100	2	
59	59		Lidia	Zajac	789654987	Dostawca	3 050	2	
60	60		Michał	Gajewski	321654789	Dostawca	3 150	2	
61	61		Nina	Dębowska	456789987	Dostawca	3 000	2	
62	62		Oliwier	Pietrzak	654987123	Dostawca	3 200	2	
63	63		Piotr	Mazowiecki	789123987	Dostawca	3 100	2	
64	64		Sara	Lisowska	123654987	Dostawca	3 050	2	

j) Przesyłka

```
CREATE TABLE przesyłka (
  ID_przesyłki      int(10) NOT NULL AUTO_INCREMENT,
  waga              float NOT NULL,
  rozmiar           varchar(2) NOT NULL,
  ID_nadawcy        int(10) NOT NULL,
  ID_adresata        int(10) NOT NULL,
  ID_rachunku        int(10) NOT NULL,
  PRIMARY KEY (ID_przesyłki),
  FOREIGN KEY (ID_nadawcy) REFERENCES nadawca (ID_nadawcy),
  FOREIGN KEY (ID_adresata) REFERENCES adresat (ID_adresata),
  FOREIGN KEY (ID_rachunku) REFERENCES rachunek (ID_rachunku));
```

```
INSERT INTO przesyłka (ID_przesyłki, waga, rozmiar, ID_nadawcy,
ID_adresata, ID_rachunku) VALUES
(1, 2.5, 'M', 2, 1, 1),
(2, 1.2, 'S', 1, 7, 2),
(3, 3.8, 'L', 1, 3, 3),
(4, 0.8, 'S', 4, 4, 4),
(5, 4.2, 'XL', 5, 5, 5),
(6, 2, 'M', 6, 6, 6),
(7, 1.5, 'M', 7, 7, 7),
(8, 5, 'XL', 8, 8, 8),
(9, 0.7, 'S', 9, 9, 9),
(10, 3, 'L', 10, 10, 10),
(11, 2.2, 'M', 11, 11, 11),
(12, 0.9, 'S', 12, 12, 12),
(13, 4.5, 'XL', 13, 13, 13),
(14, 1, 'S', 14, 14, 14),
(15, 3.3, 'L', 14, 8, 15);
```

ID_przesyłki	waga	rozmiar	ID_nadawcy	ID_adresata	ID_rachunku
1	2,5	M	2	1	1
2	1,2	S	1	7	2
3	3,8	L	1	3	3
4	0,8	S	4	4	4
5	4,2	XL	5	5	5
6	2	M	6	6	6
7	1,5	M	7	7	7
8	5	XL	8	8	8
9	0,7	S	9	9	9
10	3	L	10	10	10
11	2,2	M	11	11	11
12	0,9	S	12	12	12
13	4,5	XL	13	13	13
14	1	S	14	14	14
15	3,3	L	14	8	15

k) Rachunek

```
CREATE TABLE rachunek (
  ID_rachunku      int(10) NOT NULL AUTO_INCREMENT,
  data_wystawienia datetime NOT NULL,
  kwota            float NOT NULL,
  forma_platnosci  varchar(255) NOT NULL,
  status_platnosci varchar(255) NOT NULL,
  data_oplacenia   datetime NULL,
  PRIMARY KEY (ID_rachunku));
```

```
INSERT INTO rachunek (ID_rachunku, data_wystawienia, kwota,
forma_platnosci, status_platnosci, data_oplacenia) VALUES
(1, '2024-11-03 12:30:00', 12.5, 'BLIK', 'opłacona', '2024-11-03
12:59:30'),
(2, '2024-11-03 09:00:00', 9.5, 'przelew', 'opłacona', '2024-11-04
10:00:09'),
(3, '2024-11-03 20:00:00', 19, 'karta', 'opłacona', '2024-11-03
20:30:00'),
(4, '2024-11-03 21:15:00', 9.5, 'BLIK', 'opłacona', '2024-11-03
21:21:21'),
(5, '2024-11-03 21:00:00', 28.5, 'przelew', 'opłacona', '2024-11-04
10:10:10'),
(6, '2024-11-03 23:00:00', 12.5, 'karta', 'opłacona', '2024-11-03
23:23:23'),
(7, '2024-11-03 20:00:00', 12.5, 'BLIK', 'opłacona', '2024-11-03
20:20:20'),
(8, '2024-11-03 19:00:00', 28.5, 'BLIK', 'opłacona', '2024-11-03
19:19:19'),
(9, '2024-11-03 18:00:00', 9.5, 'BLIK', 'opłacona', '2024-11-03
18:18:18'),
(10, '2024-11-03 20:00:00', 19, 'przelew', 'opłacona', '2024-11-04
10:15:10'),
(11, '2024-11-04 20:20:00', 12.5, 'przelew', 'nieopłacona', NULL),
(12, '2024-11-04 09:00:00', 9.5, 'przelew', 'nieopłacona', NULL),
(13, '2024-11-04 15:00:00', 28.5, 'przelew', 'nieopłacona', NULL),
(14, '2024-11-04 16:00:00', 9.5, 'przelew', 'nieopłacona', NULL),
(15, '2024-11-04 17:00:00', 19, 'przelew', 'nieopłacona', NULL);
```

	ID_rachunku	data_wystawienia	kwota	forma_platnosci	status_platnosci	data_oplacenia
1	1	2024-11-03 12:30:00	12,5	BLIK	opłacona	2024-11-03 12:59:30
2	2	2024-11-03 09:00:00	9,5	przelew	opłacona	2024-11-04 10:00:09
3	3	2024-11-03 20:00:00	19	karta	opłacona	2024-11-03 20:30:00
4	4	2024-11-03 21:15:00	9,5	BLIK	opłacona	2024-11-03 21:21:21
5	5	2024-11-03 21:00:00	28,5	przelew	opłacona	2024-11-04 10:10:10
6	6	2024-11-03 23:00:00	12,5	karta	opłacona	2024-11-03 23:23:23
7	7	2024-11-03 20:00:00	12,5	BLIK	opłacona	2024-11-03 20:20:20
8	8	2024-11-03 19:00:00	28,5	BLIK	opłacona	2024-11-03 19:19:19
9	9	2024-11-03 18:00:00	9,5	BLIK	opłacona	2024-11-03 18:18:18
10	10	2024-11-03 20:00:00	19	przelew	opłacona	2024-11-04 10:15:10
11	11	2024-11-04 20:20:00	12,5	przelew	nieopłacona	(NULL)
12	12	2024-11-04 09:00:00	9,5	przelew	nieopłacona	(NULL)
13	13	2024-11-04 15:00:00	28,5	przelew	nieopłacona	(NULL)
14	14	2024-11-04 16:00:00	9,5	przelew	nieopłacona	(NULL)
15	15	2024-11-04 17:00:00	19	przelew	nieopłacona	(NULL)

l) Realizacja dostawy

```
CREATE TABLE realizacja_dostawy (
  data_przypisania datetime NOT NULL,
  ID_kuriera        int(10) NOT NULL,
  ID_przesyłki      int(10) NOT NULL,
  data_zakonczenia datetime NULL),
FOREIGN KEY (ID_kuriera) REFERENCES kurier (ID_kuriera),
FOREIGN KEY (ID_przesyłki) REFERENCES przesyłka (ID_przesyłki));
```


```
INSERT INTO realizacja_dostawy (data_przypisania, ID_kuriera, ID_przesyłki,
data_zakonczenia) VALUES
('2024-11-03 21:21:21', 1, 4, '2024-11-04 15:35:32'),
('2024-11-03 18:18:18', 2, 9, '2024-11-04 13:10:04'),
('2024-11-03 19:19:19', 3, 8, '2024-11-04 14:55:35'),
('2024-11-03 12:59:30', 4, 1, '2024-11-04 15:19:54'),
('2024-11-03 20:30:00', 7, 3, '2024-11-04 14:58:54'),
('2024-11-04 10:00:09', 7, 2, '2024-11-05 15:20:29'),
('2024-11-04 10:15:10', 12, 10, '2024-11-05 13:39:13'),
('2024-11-04 10:10:10', 15, 5, '2024-11-05 13:10:02'),
('2024-11-03 20:20:20', 16, 7, '2024-11-04 14:12:27'),
('2024-11-03 23:23:23', 16, 6, '2024-11-04 15:44:21'),
('2024-11-04 20:23:23', 2, 6, '2024-11-05 09:17:40'),
('2024-11-04 20:12:30', 2, 4, '2024-11-05 11:49:30'),
('2024-11-05 21:10:10', 6, 5, '2024-11-06 10:01:55'),
('2024-11-04 21:30:00', 6, 3, '2024-11-05 12:33:27'),
('2024-11-04 20:40:00', 7, 7, '2024-11-05 10:02:22'),
('2024-11-05 20:00:09', 7, 2, '2024-11-06 11:19:58'),
('2024-11-04 20:59:30', 7, 1, '2024-11-05 09:52:29'),
('2024-11-04 21:18:30', 12, 9, '2024-11-05 11:26:49'),
('2024-11-04 19:59:00', 14, 8, '2024-11-05 11:04:14');
```

data_przypisania	ID_kuriera	ID_przesyłki	data_zakonczenia
2024-11-03 12:59:30	4	1	2024-11-04 15:19:54
2024-11-04 20:59:30	7	1	2024-11-05 09:52:29
2024-11-04 10:00:09	7	2	2024-11-05 15:20:29
2024-11-05 20:00:09	7	2	2024-11-06 11:19:58
2024-11-03 20:30:00	7	3	2024-11-04 14:58:54
2024-11-04 21:30:00	6	3	2024-11-05 12:33:27
2024-11-03 21:21:21	1	4	2024-11-04 15:35:32
2024-11-04 20:12:30	2	4	2024-11-05 11:49:30
2024-11-04 10:10:10	15	5	2024-11-05 13:10:02
2024-11-05 21:10:10	6	5	2024-11-06 10:01:55
2024-11-03 23:23:23	16	6	2024-11-04 15:44:21
2024-11-04 20:23:23	2	6	2024-11-05 09:17:40
2024-11-03 20:20:20	16	7	2024-11-04 14:12:27
2024-11-04 20:40:00	7	7	2024-11-05 10:02:22
2024-11-03 19:19:19	3	8	2024-11-04 14:55:35
2024-11-04 19:59:00	14	8	2024-11-05 11:04:14
2024-11-03 18:18:18	2	9	2024-11-04 13:10:04
2024-11-04 21:18:30	12	9	2024-11-05 11:26:49
2024-11-04 10:15:10	12	10	2024-11-05 13:39:13

m) Stan przesyłki

```
CREATE TABLE stan_przesyłki (
  ID_stanu int(10) NOT NULL AUTO_INCREMENT,
  stan      varchar(255) NOT NULL,
  opis      varchar(255) NOT NULL,
  PRIMARY KEY (ID_stanu));
```

```
INSERT INTO stan_przesyłki (ID_stanu, stan, opis) VALUES
  (1, 'Oczekuje na płatność', 'Przesyłka oczekuje na opłacenie'),
  (2, 'Oczekuje na wysyłkę', 'Przesyłka jest opłacona i czeka na
wysłanie'),
  (3, 'Wysłana', 'Przesyłka trafiła w ręce kuriera i jedzie do
magazynu'),
  (4, 'Uszkodzona', 'Przesyłka jest zgłoszona jako uszkodzona'),
  (5, 'W magazynie', 'Przesyłka znajduje się w magazynie.'),
  (6, 'Zgubiona', 'Przesyłka została zgubiona'),
  (7, 'Wstrzymana', 'Przesyłka jest opóźniona lub wstrzymana'),
  (8, 'W punkcie dystrybucyjnym', 'Przesyłka znajduje się w punkcie
dystrybucyjnym'),
  (9, 'Reklamacja', 'Złożono reklamację na przesyłkę'),
  (10, 'Anulowana', 'Zamówienie zostało anulowane'),
  (11, 'Nieprzyjęta przez adresata', 'Przesyłka nie została przyjęta
przez adresata'),
  (12, 'W drodze do adresata', 'Przesyłka jest w drodze do adresata'),
  (13, 'Zwrócona', 'Przesyłka została zwrócona do nadawcy'),
  (14, 'W drodze do nadawcy', 'Przesyłka jest w drodze do nadawcy
(zwrot)'),
  (15, 'Dostarczona', 'Przesyłka została dostarczona do odbiorcy');
```

	ID_stanu 	stan	opis
1	1	Oczekuje na płatność	Przesyłka oczekuje na opłacenie
2	2	Oczekuje na wysyłkę	Przesyłka jest opłacona i czeka na wysłanie
3	3	Wysłana	Przesyłka trafiła w ręce kuriera i jedzie do magazynu
4	4	Uszkodzona	Przesyłka jest zgłoszona jako uszkodzona
5	5	W magazynie	Przesyłka znajduje się w magazynie.
6	6	Zgubiona	Przesyłka została zgubiona
7	7	Wstrzymana	Przesyłka jest opóźniona lub wstrzymana
8	8	W punkcie dystrybucyjnym	Przesyłka znajduje się w punkcie dystrybucyjnym
9	9	Reklamacja	Złożono reklamację na przesyłkę
10	10	Anulowana	Zamówienie zostało anulowane
11	11	Nieprzyjęta przez adresata	Przesyłka nie została przyjęta przez adresata
12	12	W drodze do adresata	Przesyłka jest w drodze do adresata
13	13	Zwrócona	Przesyłka została zwrócona do nadawcy
14	14	W drodze do nadawcy	Przesyłka jest w drodze do nadawcy (zwrot)
15	15	Dostarczona	Przesyłka została dostarczona do odbiorcy

n) Uprawnienia

```
CREATE TABLE uprawnienia (
    uprawnienia    varchar(255),
    ID_magazyniera int(10) NOT NULL,
    FOREIGN KEY (ID_magazyniera) REFERENCES magazynier (ID_magazyniera));
```

```
INSERT INTO uprawnienia (ID_magazyniera, uprawnienia) VALUES
    (1, 'Administrator systemu'),
    (2, 'Generowanie raportow'),
    (2, 'Podglad stanu magazynow'),
    (3, 'Rejestracja zwrotow'),
    (4, 'Przyjmowanie dostaw'),
    (5, 'Przyjmowanie dostaw'),
    (5, 'Podglad stanu magazynow'),
    (6, 'Administrator systemu'),
    (7, 'Rejestracja zwrotow'),
    (7, 'Przyjmowanie dostaw'),
    (8, 'Generowanie raportow'),
    (9, 'Przyjmowanie dostaw'),
    (10, 'Administrator systemu'),
    (11, 'Zarzadzanie dostawami'),
    (11, 'Generowanie raportow');
```

	uprawnienia	ID_magazyniera 
1	Administrator systemu	1
2	Generowanie raportow	2
3	Podglad stanu magazynow	2
4	Rejestracja zwrotow	3
5	Przyjmowanie dostaw	4
6	Przyjmowanie dostaw	5
7	Podglad stanu magazynow	5
8	Administrator systemu	6
9	Rejestracja zwrotow	7
10	Przyjmowanie dostaw	7
11	Generowanie raportow	8
12	Przyjmowanie dostaw	9
13	Administrator systemu	10
14	Zarzadzanie dostawami	11
15	Generowanie raportow	11

Zapytania SQL

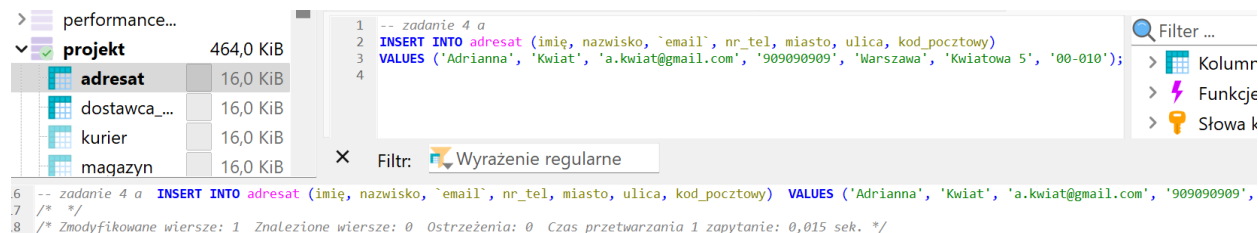
a) Dopisywanie danych do bazy – INSERT INTO

Zapytanie do dodawania informacji do bazy np. dodanie nowego adresata

```
INSERT INTO adresat (imię, nazwisko, `email`, nr_tel, miasto, ulica, kod_pocztowy)
VALUES ('Adrianna', 'Kwiat', 'a.kwiat@gmail.com', '909090909', 'Warszawa', 'Kwiatowa 5', '00-010');
```

Jak chcemy dodać więcej wartości naraz to:

```
INSERT INTO adresat (imię, nazwisko, `email`, nr_tel, miasto, ulica, kod_pocztowy)
VALUES ('Adrianna', 'Kwiat', 'a.kwiat@gmail.com', '909090909', 'Warszawa', 'Kwiatowa 5', '00-010'),
('Adam', 'Kosa', 'a.kosa@gmail.com', '707070707', 'Lublin', 'Rubinowa 8', '23-651');
```



#	ID_adresata	imię	nazwisko	email	nr_tel	miasto	ulica	kod_pocztowy
1	1	Jan	Kowalski	j.kowalski@gmail.com	123456789	Warszawa	Ulica 1	00-001
2	2	Anna	Nowak	a.nowak@gmail.com	234567890	Warszawa	Ulica 2	00-015
3	3	Marek	Zielinski	m.zielinski@gmail.com	345678901	Kraków	Ulica 3	30-001
4	4	Kasia	Wojcik	k.wojcik@gmail.com	456789012	Bydgoszcz	Ulica 4	85-001
5	5	Piotr	Wisnia	p.wisnia@gmail.com	567890123	Kraków	Ulica 5	30-020
6	6	Ewa	Kaczmarek	e.kaczmarek@gmail.com	678901234	Bydgoszcz	Ulica 6	85-010
7	7	Tomasz	Dab	t.dab@gmail.com	789012345	Warszawa	Ulica 7	01-234
8	8	Agnieszka	Aga	a.ag@gmail.com	890123456	Olsztyn	Ulica 8	10-001
9	9	Rafal	Nowak	r.nowak@gmail.com	901234567	Katowice	Ulica 9	40-001
10	10	Justyna	Kowalczyk	j.kowalczyk@gmail.com	987654321	Szczecin	Ulica 10	70-001
11	11	Jacek	Maj	j.maj@gmail.com	876543210	Opole	Ulica 11	45-001
12	12	Martyna	Mazur	m.mazur@gmail.com	765432109	Bydgoszcz	Ulica 12	85-020
13	13	Jacek	Wrobel	j.wrobel@gmail.com	654321098	Lublin	Ulica 13	20-001
14	14	Szymon	Pawlak	s.pawlak@gmail.com	543210987	Kraków	Ulica 14	30-030
15	15	Natalia	Szymczak	n.szymczak@gmail.com	432109876	Rzeszów	Ulica 15	35-001
16	16	Adrianna	Kwiat	a.kwiat@gmail.com	909090909	Warszawa	Kwiatowa 5	00-010

b) Modyfikowanie danych w bazie – UPDATE

Modyfikacja danych w tabeli np. adresat:

```
UPDATE adresat
SET miasto = 'Kraków', ulica = 'Ogrodowa 10'
WHERE ID_adresata = 16;
```

performance...
projekt 464,0 KiB
adresat 16,0 KiB
dostawca_... 16,0 KiB

1 UPDATE adresat
2 SET miasto = 'Kraków', ulica = 'Ogrodowa 10'
3 WHERE ID_adresata = 16;
4

Filtr: Wyrażenie regularne

UPDATE adresat SET miasto = 'Kraków', ulica = 'Ogrodowa 10' WHERE ID_adresata = 16;
/* */
/* Info: Rows matched: 1 Changed: 1 Warnings: 0 */
/* Zmodyfikowane wiersze: 1 Znalezione wiersze: 0 Ostrzeżenia: 0 Czas przetwarzania 1 zapytanie: 0,047 sek. */

#	ID_adresata	imię	nazwisko	email	nr_tel	miasto	ulica	kod_pocztowy
1	1	Jan	Kowalski	j.kowalski@gmail.com	123456789	Warszawa	Ulica 1	00-001
2	2	Anna	Nowak	a.nowak@gmail.com	234567890	Warszawa	Ulica 2	00-015
3	3	Marek	Zielinski	m.zielinski@gmail.com	345678901	Kraków	Ulica 3	30-001
4	4	Kasia	Wojcik	k.wojcik@gmail.com	456789012	Bydgoszcz	Ulica 4	85-001
5	5	Piotr	Wisnia	p.wisnia@gmail.com	567890123	Kraków	Ulica 5	30-020
6	6	Ewa	Kaczmarek	e.kaczmarek@gmail.com	678901234	Bydgoszcz	Ulica 6	85-010
7	7	Tomasz	Dab	t.dab@gmail.com	789012345	Warszawa	Ulica 7	01-234
8	8	Agnieszka	Aga	a.ag@gmail.com	890123456	Olsztyn	Ulica 8	10-001
9	9	Rafal	Nowak	r.nowak@gmail.com	901234567	Katowice	Ulica 9	40-001
10	10	Justyna	Kowalczyk	j.kowalczyk@gmail.com	987654321	Szczecin	Ulica 10	70-001
11	11	Jacek	Maj	j.maj@gmail.com	876543210	Opole	Ulica 11	45-001
12	12	Martyna	Mazur	m.mazur@gmail.com	765432109	Bydgoszcz	Ulica 12	85-020
13	13	Jacek	Wrobel	j.wrobel@gmail.com	654321098	Lublin	Ulica 13	20-001
14	14	Szymon	Pawlak	s.pawlak@gmail.com	543210987	Kraków	Ulica 14	30-030
15	15	Natalia	Szymczak	n.szymczak@gmail.com	432109876	Rzeszów	Ulica 15	35-001
16	16	Adrianna	Kwiat	a.kwiat@gmail.com	909090909	Kraków	Ogrodowa 10	00-010

c) Usuwanie danych z bazy – DELETE

Usuwanie danych np. konkretnego adresata:

```
DELETE FROM adresat
WHERE ID_adresata = 16;
```

```
DELETE FROM adresat WHERE ID_adresata = 16;  
/* */  
/* Zmodyfikowane wiersze: 1 Znalezione wiersze: 0 Ostrzeżenia: 0 Czas przetwarzania 1 zapytanie: 0,016 sek. */
```

	ID_adresata	imię	nazwisko	email	nr_tel	miasto	ulica	kod_pocztowy
1	1	Jan	Kowalski	j.kowalski@gmail.com	123456789	Warszawa	Ulica 1	00-001
2	2	Anna	Nowak	a.nowak@gmail.com	234567890	Warszawa	Ulica 2	00-015
3	3	Marek	Zielinski	m.zielinski@gmail.com	345678901	Kraków	Ulica 3	30-001
4	4	Kasia	Wojcik	k.wojcik@gmail.com	456789012	Bydgoszcz	Ulica 4	85-001
5	5	Piotr	Wisnia	p.wisnia@gmail.com	567890123	Kraków	Ulica 5	30-020
6	6	Ewa	Kaczmarek	e.kaczmarek@gmail.com	678901234	Bydgoszcz	Ulica 6	85-010
7	7	Tomasz	Dab	t.dab@gmail.com	789012345	Warszawa	Ulica 7	01-234
8	8	Agnieszka	Aga	a.ag@gmail.com	890123456	Olsztyn	Ulica 8	10-001
9	9	Rafal	Nowak	r.nowak@gmail.com	901234567	Katowice	Ulica 9	40-001
10	10	Justyna	Kowalczyk	j.kowalczyk@gmail.com	987654321	Szczecin	Ulica 10	70-001
11	11	Jacek	Maj	j.maj@gmail.com	876543210	Opole	Ulica 11	45-001
12	12	Martyna	Mazur	m.mazur@gmail.com	765432109	Bydgoszcz	Ulica 12	85-020
13	13	Jacek	Wrobel	j.wrobel@gmail.com	654321098	Lublin	Ulica 13	20-001
14	14	Szymon	Pawlak	s.pawlak@gmail.com	543210987	Kraków	Ulica 14	30-030
15	15	Natalia	Szymczak	n.szymczak@gmail.com	432109876	Rzeszów	Ulica 15	35-001

d) Pobieranie danych z bazy – SELECT

1) SELECT – wyświetlenie danych

Zapytanie służące do wyświetlenia wszystkich kolumn z tabeli np. magazyn:

```
SELECT * FROM magazyn
```

```
1 SELECT * FROM magazyn
```

magazyn (17r × 6c)						
	ID_magazynu	miasto	ulica	pojemnosc	kod_pocztyowy	województwo
1	1	Warszawa	Magazynowa 1	1 000	00-030	Mazowieckie
2	2	Kraków	Magazynowa 2	800	30-004	Małopolskie
3	3	Gdańsk	Magazynowa 3	1 200	80-009	Pomorskie
4	4	Wrocław	Magazynowa 4	900	50-010	Dolnośląskie
5	5	Poznań	Magazynowa 5	1 100	60-059	Wielkopolskie
6	6	Szczecin	Magazynowa 6	950	70-002	Zachodnio-Pomorskie
7	7	Lublin	Magazynowa 7	1 300	20-033	Lubelskie
8	8	Bydgoszcz	Magazynowa 8	1 000	85-070	Kujawsko-Pomorskie
9	9	Katowice	Magazynowa 9	850	40-022	Śląskie
10	10	Białystok	Magazynowa 10	1 200	15-001	Podlaskie
11	11	Olsztyn	Magazynowa 11	950	10-009	Warmińsko-Mazurskie
12	12	Rzeszów	Magazynowa 12	1 100	35-007	Podkarpackie
13	13	Zielona Góra	Magazynowa 13	900	65-111	Lubuskie
14	14	Opole	Magazynowa 14	1 150	45-003	Opolskie
15	15	Łódź	Magazynowa 15	1 000	90-005	Łódzkie
16	16	Kielce	Magazynowa 16	1 050	25-002	Świętokrzyskie
17	17	Piotrków Trybunalski	Dystrybucyjna 1	18 000	97-300	Łódzkie

2) Złączenie wewnętrzne – INNER JOIN

Zapytanie, które wyświetla pracowników i pojazdy, które są do nich przypisane:

```
SELECT pracownik.ID_pracownika, pracownik.imię, pracownik.nazwisko,  
pojazd.typ_pojazdu, pojazd.nr_rejestracyjny  
FROM pracownik  
INNER JOIN pojazd ON pracownik.ID_pracownika = pojazd.ID_pracownika;
```

Wynik #1 (42r × 5c)					
#	ID_pracownika	imię	nazwisko	typ_pojazdu	nr_rejestracyjny
1	4	Dawid	Wojcik	Bus	ZXC12345
2	6	Filip	Kowalczyk	Bus	RTY45678
3	8	Hanna	Mazur	Samochód dostawczy	YUI78901
4	10	Jan	Olszewski	Bus	QWE12345
5	12	Lukasz	Pacek	Bus	ASD45678
6	14	Natalia	Szymczak	Samochód dostawczy	FGH78901
7	16	Piotr	Bielecki	Bus	HJK12345
8	22	Zofia	Król	Van	BNJ45678
9	24	Jakub	Jankowski	Bus	MNB78901
10	26	Barbara	Turek	Bus	VFR12345
11	28	Magdalena	Wilk	Samochód dostawczy	TGW45678
12	30	Kacper	Pawlak	Bus	ZHN78901
13	32	Oskar	Jablonski	Bus	PLM12345
14	34	Robert	Broza	Bus	QAZ45678
15	36	Tadeusz	Stankiewicz	Samochód dostawczy	WSX78901
16	38	Zbigniew	Polak	Bus	EDC12345
17	40	Maria	Sowa	Bus	RFV45678
18	16	Piotr	Bielecki	Samochód dostawczy	TGB78901
19	41	Alicja	Kowalska	Bus	YHN12345
20	42	Bartosz	Malinowski	Van	UJM45678
21	43	Cezary	Nowicki	Bus	IKL78901
22	44	Daria	Gorska	Bus	OPZ12345
23	45	Edward	Jurek	Samochód dostawczy	XYZ45678
24	46	Franciszek	Rataj	Bus	NHG45678
25	47	Gabriela	Król	Bus	BNM78901
26	48	Hubert	Zakrzewski	Bus	LK12345
27	49	Izabela	Baran	Van	POI78901
28	50	Jacek	Dąbrowski	Bus	UYT12345
29	51	Katarzyna	Polanska	Bus	TRE45678
30	52	Leon	Ostrowski	Van	QAZ78901
31	53	Marcelina	Zięba	Van	ZXA44444
32	54	Norbert	Wieczorek	Bus	RTY55555
33	55	Olaf	Wróbel	Samochód dostawczy	YUI66666
34	56	Paulina	Kucharska	Bus	QWE77777
35	57	Ryszard	Bartosz	Bus	LJM12342
36	58	Krzysztof	Nowak	Samochód dostawczy	SAM98765
37	59	Lidia	Zajęc	Van	TGV98765
38	60	Michał	Gajewski	Bus	BNQ87654
39	61	Nina	Dębowska	Samochód dostawczy	PLM65432
40	62	Oliwier	Pietrzak	Bus	ASX54321
41	63	Piotr	Mazowiecki	Van	UJM43210
42	64	Sara	Lisowska	Bus	KLM32109

3) Złączenie zewnętrzne – LEFT OUTER JOIN

LEFT OUTER JOIN zwraca wszystkie kolumny z lewej tabeli, nawet jeśli nie mają wartości wspólnych z drugą tabelą. Analogicznie RIGHT OUTER JOIN.

Zapytanie, które wyświetla wszystkich pracowników oraz ich pojazdy:

```
SELECT pracownik.ID_pracownika, pracownik.imię, pracownik.nazwisko,
pojazd.typ_pojazdu, pojazd.nr_rejestracyjny
FROM pracownik
LEFT OUTER JOIN pojazd ON pracownik.ID_pracownika = pojazd.ID_pracownika;
```

Wynik #1 (65r x 5c)					
#	ID_pracownika	imię	nazwisko	typ_pojazdu	nr_rejestracyjny
1	1	Adam	Nowak	(NULL)	(NULL)
2	2	Bartek	Zielinski	(NULL)	(NULL)
3	3	Cezary	Adamski	(NULL)	(NULL)
4	4	Dawid	Wojcik	Bus	ZXC12345
5	5	Ewa	Sikora	(NULL)	(NULL)
6	6	Filip	Kowalczyk	Bus	RTY45678
7	7	Grzegorz	Zawadzki	(NULL)	(NULL)
8	8	Hanna	Mazur	Samochód dostawczy	YUI78901
9	9	Iwona	Czajka	(NULL)	(NULL)
10	10	Jan	Olaszewski	Bus	QWE12345
11	11	Kamil	Duda	(NULL)	(NULL)
12	12	Lukasz	Pacek	Bus	ASD45678
13	13	Marta	Szymczak	(NULL)	(NULL)
14	14	Natalia	Szymczak	Samochód dostawczy	FGH78901
15	15	Olivia	Czerwonna	(NULL)	(NULL)
16	16	Piotr	Bielecki	Bus	HJK12345
17	16	Piotr	Bielecki	Samochód dostawczy	TGB78901
18	17	Rafał	Szczepaniak	(NULL)	(NULL)
19	18	Szymon	Gorski	(NULL)	(NULL)
20	19	Tomasz	Chmielewski	(NULL)	(NULL)
21	20	Urszula	Jakubowska	(NULL)	(NULL)
22	21	Wojciech	Lis	(NULL)	(NULL)
23	22	Zofia	Król	Van	BNJ45678
24	23	Paweł	Sawicki	(NULL)	(NULL)
25	24	Jakub	Jankowski	Bus	MNB78901
26	25	Karolina	Lesniak	(NULL)	(NULL)
27	26	Barbara	Turek	Bus	VFR12345
28	27	Krzysztof	Borowski	(NULL)	(NULL)
29	28	Magdalena	Wilk	Samochód dostawczy	TGW45678
30	29	Emilia	Błaszczak	(NULL)	(NULL)
31	30	Kacper	Pawlak	Bus	ZHN78901
32	31	Olga	Kowal	(NULL)	(NULL)
33	32	Oskar	Jablonski	Bus	PLM12345
34	33	Patryk	Gajda	(NULL)	(NULL)
35	34	Robert	Broza	Bus	QAZ45678
36	35	Sylwia	Kaminska	(NULL)	(NULL)
37	36	Tadeusz	Stankiewicz	Samochód dostawczy	WSX78901
38	37	Wiktoria	Mocek	(NULL)	(NULL)
39	38	Zbigniew	Polak	Bus	EDC12345
40	39	Janusz	Zagorski	(NULL)	(NULL)
41	40	Maria	Sowa	Bus	RFV45678
42	41	Alicja	Kowalska	Bus	YHN12345
43	42	Bartosz	Malinowski	Van	UJM45678
44	43	Cezary	Nowicki	Bus	IKL78901
45	44	Daria	Gorska	Bus	OPZ12345
46	45	Edward	Jurek	Samochód dostawczy	XYZ45678
47	46	Franciszek	Rataj	Bus	NHG45678
48	47	Gabriela	Król	Bus	BNM78901
49	48	Hubert	Zakrzewski	Bus	LKI12345
50	49	Izabela	Baran	Van	POI78901
51	50	Jacek	Dąbrowski	Bus	UYT12345
52	51	Katarzyna	Polanska	Bus	TRE45678
53	52	Leon	Ostrowski	Van	QAZ78901
54	53	Marcelina	Zięba	Van	ZXA44444
55	54	Norbert	Wieczorek	Bus	RTY55555
56	55	Olaf	Wróbel	Samochód dostawczy	YUI66666
57	56	Paulina	Kucharska	Bus	QWE77777
58	57	Ryszard	Bartos	Bus	LIM12342
59	58	Krzysztof	Nowak	Samochód dostawczy	SAM98765
60	59	Lidia	Zając	Van	TGV98765
61	60	Michał	Gajewski	Bus	BNQ87654
62	61	Nina	Dębowska	Samochód dostawczy	PLM65432
63	62	Oliwier	Pietrzak	Bus	ASX54321
64	63	Piotr	Mazowiecki	Van	UJM43210
65	64	Sara	Lisowska	Bus	KLM32109

4) Złączenie zewnętrzne - RIGHT OUTER JOIN

Zapytanie, które wyświetla wszystkie pojazdy oraz pracowników, do których są przypisane te pojazdy:

```
SELECT pracownik.ID_pracownika, pracownik.imię, pracownik.nazwisko,
pojazd.typ_pojazdu, pojazd.nr_rejestracyjny
FROM pracownik
RIGHT OUTER JOIN pojazd ON pracownik.ID_pracownika = pojazd.ID_pracownika;
```

Wynik #1 (43r x 5c)

#	ID_pracownika	imię	nazwisko	typ_pojazdu	nr_rejestracyjny
1	4	Dawid	Wojcik	Bus	ZXC12345
2	6	Filip	Kowalczyk	Bus	RTY45678
3	8	Hanna	Mazur	Samochód dostawczy	YUI78901
4	10	Jan	Olszewski	Bus	QWE12345
5	12	Lukasz	Pacek	Bus	ASD45678
6	14	Natalia	Szymczak	Samochód dostawczy	FGH78901
7	16	Piotr	Bielecki	Bus	HJK12345
8	22	Zofia	Krol	Van	BNJ45678
9	24	Jakub	Jankowski	Bus	MNB78901
10	26	Barbara	Turek	Bus	VFR12345
11	28	Magdalena	Wilk	Samochód dostawczy	TGW45678
12	30	Kacper	Pawlak	Bus	ZHN78901
13	32	Oskar	Jablonski	Bus	PLM12345
14	34	Robert	Broza	Bus	QAZ45678
15	36	Tadeusz	Stankiewicz	Samochód dostawczy	WSX78901
16	38	Zbigniew	Polak	Bus	EDC12345
17	40	Maria	Sowa	Bus	RFV45678
18	16	Piotr	Bielecki	Samochód dostawczy	TGB78901
19	41	Alicja	Kowalska	Bus	YHN12345
20	42	Bartosz	Malinowski	Van	UJM45678
21	43	Cezary	Nowicki	Bus	IKL78901
22	44	Daria	Gorska	Bus	OPZ12345
23	45	Edward	Jurek	Samochód dostawczy	XYZ45678
24	46	Franciszek	Rataj	Bus	NHG45678
25	47	Gabriela	Król	Bus	BNM78901
26	48	Hubert	Zakrzewski	Bus	LKI12345
27	49	Izabela	Baran	Van	POI78901
28	50	Jacek	Dąbrowski	Bus	UYT12345
29	51	Katarzyna	Polanska	Bus	TRE45678
30	52	Leon	Ostrowski	Van	QAZ78901
31	53	Marcelina	Zięba	Van	ZXA44444
32	54	Norbert	Wieczorek	Bus	RTY55555
33	55	Olaf	Wróbel	Samochód dostawczy	YUI66666
34	56	Paulina	Kucharska	Bus	QWE77777
35	57	Ryszard	Bartosz	Bus	LIM12342
36	58	Krzysztof	Nowak	Samochód dostawczy	SAM98765
37	59	Lidia	Zając	Van	TGV98765
38	60	Michał	Gajewski	Bus	BNQ87654
39	61	Nina	Dębowska	Samochód dostawczy	PLM65432
40	62	Oliwier	Pietrzak	Bus	ASX54321
41	63	Piotr	Mazowiecki	Van	UJM43210
42	64	Sara	Lisowska	Bus	KLM32109
43	(NULL)	(NULL)	(NULL)	Samochód dostawczy	BVC21098

5) Grupowanie – GROUP BY

GROUP BY służy do zgrupowania danych i wyświetlenia np. ile razy powtarza się dana wartość w danej tabeli.

Zapytanie, które wyświetla do ilu przesyłek był przypisany dany stan przesyłki:

```
SELECT stan_przesyłki.stan, COUNT(przesylka.ID_przesyłki) AS  
liczba_przesylek  
FROM przesylka  
JOIN opis_stanu_przesylki ON przesylka.ID_przesyłki =  
opis_stanu_przesylki.ID_przesyłki  
JOIN stan_przesyłki ON opis_stanu_przesylki.ID_stanu =  
stan_przesyłki.ID_stanu  
GROUP BY stan_przesyłki.stan;
```


#	stan	liczba_przesylek
1	Oczekuje na płatność	15
2	Oczekuje na wysyłkę	10
3	Wysłana	10
4	W magazynie	18
5	W drodze do adresata	9
6	Dostarczona	9
7	W punkcie dystrybucyjnym	9

6) Sortowanie – ORDER BY

Można sortować rosnąco (ASC) i malejąco (DESC).

Zapytanie, które wyświetla 10 ID_przesyłki oraz jej wagę, posortowane malejąco według wagi:

```
SELECT ID_przesyłki, waga  
FROM przesylka  
ORDER BY waga DESC  
LIMIT 10;
```

#	ID_przesyłki		waga
1	8		5
2	13		4,5
3	5		4,2
4	3		3,8
5	15		3,3
6	10		3
7	1		2,5
8	11		2,2
9	6		2
10	7		1,5

7) Podzapytanie nieskorelowane

Podzapytanie działa niezależnie od zapytania nadrzędnego. Jest wykonywane tylko raz.
Np. Magazyny, w których przechowywane są przesyłki do Warszawy:

```
SELECT DISTINCT magazyn.ID_magazynu, magazyn.miasto
FROM magazyn
WHERE magazyn.ID_magazynu IN (
    SELECT opis_stanu_przesylki.ID_magazynu
    FROM opis_stanu_przesylki
    JOIN przesyłka ON opis_stanu_przesylki.ID_przesyłki =
przesyłka.ID_przesyłki
    JOIN adresat ON przesyłka.ID_adresata = adresat.ID_adresata
    WHERE adresat.miasto = 'Warszawa');
```

magazyn (4r × 2c)		
#	ID_magazynu	miasto
1	1	Warszawa
2	2	Kraków
3	6	Szczecin
4	17	Piotrków Trybunalski

8) Podzapytanie skorelowane

Podzapytanie odwołuje się do zewnętrznych danych, które zmieniają się w zależności od wiersza zewnętrznego zapytania.

Np. Poniższe podzapytanie sprawdza czy dla każdej przesyłki istnieje odpowiednia realizacja dostawy, która spełnia określony warunek (czy kod pocztowy nadawcy mieści się w zakresie kodów pocztowych przypisanych kurierowi). Jeśli spełnia ten warunek to zapytanie zewnętrzne zwróci dane przesyłki:

```
SELECT przesyłka.ID_przesyłki, przesyłka.waga, przesyłka.rozmiar
FROM przesyłka
WHERE EXISTS (
    SELECT 1
    FROM realizacja_dostawy
    JOIN kurier ON realizacja_dostawy.ID_kuriera = kurier.ID_kuriera
    JOIN nadawca ON przesyłka.ID_nadawcy = nadawca.ID_nadawcy
    WHERE realizacja_dostawy.ID_przesyłki = przesyłka.ID_przesyłki
    AND nadawca.kod_pocztowy BETWEEN kurier.kod_pocztowy_od AND
kurier.kod_pocztowy_do
);
```

przesyłka (10r × 3c)			
#	ID_przesyłki	waga	rozmiar
1	4	0,8	S
2	9	0,7	S
3	8	5	XL
4	1	2,5	M
5	3	3,8	L
6	2	1,2	S
7	10	3	L
8	5	4,2	XL
9	7	1,5	M
10	6	2	M

9) Klauzula HAVING

Podobne do WHERE, służy do filtrowania grupy po GROUP BY:

Zapytanie, które wyświetla tylko tych nadawców, którzy wysłali więcej niż 1 przesyłkę:

```
SELECT nadawca.ID_nadawcy, nadawca.imię, nadawca.nazwisko,  
COUNT(przesylka.ID_przesyłki) AS liczba_przesyłek  
FROM nadawca  
JOIN przesylka ON nadawca.ID_nadawcy = przesylka.ID_nadawcy  
GROUP BY nadawca.ID_nadawcy  
HAVING COUNT(przesylka.ID_przesyłki) > 1;
```

#	ID_nadawcy	imię	nazwisko	liczba_przesyłek
1	1	Adam	Adamowski	2
2	14	Natalia	Nowakowska	2

10) Klauzula LIKE

LIKE – używana w WHERE do wyszukiwania określonego ciągu znaków w kolumnie.

Zapytanie służące do wyświetlenia przesyłek o rozmiarze S:

```
SELECT ID_przesyłki, waga, rozmiar  
FROM przesylka  
WHERE rozmiar LIKE 'S';
```

#	ID_przesyłki	waga	rozmiar
1	2	1,2	S
2	4	0,8	S
3	9	0,7	S
4	12	0,9	S
5	14	1	S

11) Klauzula IN

Zapytanie służące do wyświetlenia kurierów, których obszar działania to Mazowieckie, Kujawsko-Pomorskie lub Lubuskie:

```
SELECT ID_kuriera, obszar_działania  
FROM kurier  
WHERE obszar_działania IN ('Mazowieckie', 'Kujawsko-Pomorskie',  
'Lubuskie');
```

	ID_kuriera	obszar_działania
1	2	Kujawsko-Pomorskie
2	4	Lubuskie
3	7	Mazowieckie

12) Klauzula ANY

ANY umożliwia porównanie wartości z jakimś zbiorem wartości w podzapytaniu.

Zapytanie, które wyświetla pojazdy, które zostały przypisane do kurierów:

```
SELECT p.ID_pojazdu, p.typ_pojazdu
FROM pojazd p
WHERE p.ID_pracownika = ANY (
    SELECT ID_pracownika
    FROM pracownik
    WHERE stanowisko = 'Kurier'
);
```

#	ID_pojazdu	typ_pojazdu
1	1	Bus
2	2	Bus
3	3	Samochód dostawczy
4	4	Bus
5	5	Bus
6	6	Samochód dostawczy
7	7	Bus
8	18	Samochód dostawczy
9	8	Van
10	9	Bus
11	10	Bus
12	11	Samochód dostawczy
13	12	Bus
14	13	Bus
15	14	Bus
16	15	Samochód dostawczy
17	16	Bus
18	17	Bus

13) Klauzula ALL

Zapytanie, które wyświetla pracowników, których pensja jest większa od pensji wszystkich kurierów:

```
SELECT imię, nazwisko, pensja
FROM pracownik
WHERE pensja > ALL (
    SELECT pensja
    FROM pracownik
    WHERE stanowisko = 'Kurier'
);
```

#	imię	nazwisko	pensja
1	Adam	Nowak	12 000
2	Bartek	Zielinski	5 200
3	Cezary	Adamski	5 200
4	Kamil	Duda	3 300
5	Szymon	Gorski	3 300
6	Tomasz	Chmielewski	3 400
7	Urszula	Jakubowska	3 450
8	Wojciech	Lis	3 500

14) Klauzula EXISTS

Zapytanie, które wyświetla nadawców, którzy mają nieopłacone przesyłki.

```
SELECT n.imię, n.nazwisko
FROM nadawca n
WHERE EXISTS (
    SELECT 1
    FROM przesyłka p
    JOIN rachunek r ON p.ID_rachunku = r.ID_rachunku
    WHERE p.ID_nadawcy = n.ID_nadawcy
    AND r.status_platnosci = 'nieopłacona'
);
```

nadawca (4r × 2c)		
#	imię	nazwisko
1	Krzysztof	Krysiak
2	Lidia	Lidzia
3	Marek	Marekowski
4	Natalia	Nowakowska

15) Dodatkowe zapytanie

Zapytanie, które wyświetla ID_przesyłki i ID_kuriera, który działa w obszarze nadawcy
wybranych przesyłek (1-5):

```
SELECT
    przesyłka.ID_przesyłki,
    kurier.kod_pocztowy_od,
    kurier.kod_pocztowy_do,
    nadawca.kod_pocztowy AS kod_pocztowy_nadawcy,
    kurier.ID_kuriera
FROM
    przesyłka
JOIN
    nadawca ON przesyłka.ID_nadawcy = nadawca.ID_nadawcy
JOIN
    kurier ON nadawca.kod_pocztowy BETWEEN kurier.kod_pocztowy_od AND
    kurier.kod_pocztowy_do
WHERE
    przesyłka.ID_przesyłki IN (1, 2, 3, 4, 5);
```

	ID_przesyłki		kod_pocztowy_od	kod_pocztowy_do	kod_pocztowy_nadawcy	ID_kuriera
1	4		50-000	59-999	50-003	1
2	1		30-000	34-999	31-010	6
3	3		00-000	09-999	00-001	7
4	2		00-000	09-999	00-001	7
5	5		60-000	64-999	60-100	15

Procedury składowane

- 1) Procedura, która przypisuje kuriera do nadawcy – jeśli nie ma żadnych wpisów odnośnie przesyłki, do której chcemy przypisać kuriera w tabeli 'realizacja_dostawy', dodaje się wiersz do tej tabeli z odpowiednim ID_kuriera (ID kuriera, który obsługuje kod pocztowy nadawcy).

```
DELIMITER $$
CREATE PROCEDURE przypisz_kuriera_do_nadawcy (IN p_kod_pocztowy_nadawcy
VARCHAR(6), IN p_ID_przesyłki INT)
BEGIN
    DECLARE v_ID_kuriera INT;
    DECLARE v_count INT;

    -- Wyszukaj kuriera, który obsługuje dany kod pocztowy nadawcy
    SELECT ID_kuriera
    INTO v_ID_kuriera
    FROM kurier
    WHERE p_kod_pocztowy_nadawcy BETWEEN kod_pocztowy_od AND
kod_pocztowy_do
    LIMIT 1;
    -- Jeśli kurier został znaleziony
    IF v_ID_kuriera IS NOT NULL THEN
        -- Sprawdź, czy przesyłka już ma przypisanego kuriera w tabeli
        realizacja_dostaw

        SELECT COUNT(*)
        INTO v_count
        FROM realizacja_dostawy
        WHERE ID_przesyłki = p_ID_przesyłki
        AND ID_kuriera IS NOT NULL;

        -- Jeśli przesyłka nie ma przypisanego kuriera (v_count = 0), dodaj
        nowy wpis
        IF v_count = 0 THEN
            INSERT INTO realizacja_dostawy (data_przypisania, ID_kuriera,
ID_przesyłki)
            VALUES (NOW(), v_ID_kuriera, p_ID_przesyłki);
        ELSE
            -- Jeśli przesyłka już ma przypisanego kuriera, zgłoś błąd
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Przesyłka ma już
przypisanego kuriera.';
        END IF;
    ELSE
        -- Jeśli nie znaleziono kuriera dla podanego kodu pocztowego
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Nie znaleziono kuriera
dla podanego kodu pocztowego.';
    END IF;
END$$
DELIMITER ;
```

Do wywołania procedury należy wprowadzić kod pocztowy nadawcy i ID_przesyłki, np.:

```
CALL przypisz_kuriera_do_nadawcy('12-345', 11);
```

- 2) Procedura, która przypisuje kuriera do adresata – jeśli przesyłka miała już przypisanego jednego kuriera (do nadawcy) oraz znaleziono kuriera, który obsługuje kod pocztowy adresata to dodaje się wiersz do tabeli 'realizacja_dostawy' z ID_kuriera, który dostarczy przesyłkę do adresata.

```
DELIMITER $$
CREATE PROCEDURE przypisz_kuriera_do_adresata ((IN p_kod_pocztowy_adresata
VARCHAR(6), IN p_ID_przesyłki INT)
BEGIN
    DECLARE v_ID_kuriera INT;
    DECLARE v_count INT;

    -- Wyszukaj kuriera, który obsługuje dany kod pocztowy adresata
    SELECT ID_kuriera
    INTO v_ID_kuriera
    FROM kurier
    WHERE p_kod_pocztowy_adresata BETWEEN kod_pocztowy_od AND
kod_pocztowy_do
    LIMIT 1;

    -- Jeśli kurier został znaleziony
    IF v_ID_kuriera IS NOT NULL THEN
        -- Sprawdź, czy przesyłka już ma przypisanego kuriera w tabeli
        realizacja_dostaw

        SELECT COUNT(*)
        INTO v_count
        FROM realizacja_dostawy
        WHERE ID_przesyłki = p_ID_przesyłki
        AND ID_kuriera IS NOT NULL;

        -- Jeśli przesyłka miała już wcześniej przypisanego jednego kuriera
        (v_count = 1), dodaj nowy wpis
        IF v_count = 1 THEN
            INSERT INTO realizacja_dostawy (data_przypisania, ID_kuriera,
ID_przesyłki)
            VALUES (NOW(), v_ID_kuriera, p_ID_przesyłki);
        ELSE
            -- Jeśli przesyłka już ma przypisanego drugiego kuriera, zgłoś
            błąd
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Przesyłka ma już
przypisanego kuriera.';
            END IF;
        ELSE
            -- Jeśli nie znaleziono kuriera dla podanego kodu pocztowego
            SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Nie znaleziono kuriera
dla podanego kodu pocztowego.';
            END IF;
    END$$
DELIMITER ;
```

Do wywołania procedury należy wprowadzić kod pocztowy adresata i ID_przesyłki, np.:

```
CALL przypisz_kuriera_do_adresata('12-345', 10);
```

- 3) Procedura dodania nowej przesyłki – najpierw sprawdza czy wprowadzeni nadawca oraz adresat istnieją, potem na podstawie rozmiaru paczki wyznaczana jest kwota, która zostanie dodana do tabeli *rachunek*. Następnie dodane są wiersze do tabel *rachunek*, *przesylka* i *opis_stanu_przesylki*, w tym początkowy stan przesyłki: „oczekuje na opłacenie”.

```
DELIMITER $$
CREATE PROCEDURE Dodaj_Nowa_Przesylke(
    IN p_ID_nadawcy INT,
    IN p_ID_adresata INT,
    IN p_waga FLOAT,
    IN p_rozmiar VARCHAR(2),
    IN p_forma_platnosci VARCHAR(255)
)
BEGIN
    -- Sprawdzenie istnienia nadawcy
    IF NOT EXISTS (SELECT 1 FROM nadawca WHERE ID_nadawcy = p_ID_nadawcy)
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Taki nadawca nie istnieje, dodaj wartości do
tabeli albo podaj istniejącego nadawcę';
    END IF;
    -- Sprawdzenie istnienia adresata
    IF NOT EXISTS (SELECT 1 FROM adresat WHERE ID_adresata = p_ID_adresata)
    THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Taki adresat nie istnieje, dodaj wartości do
tabeli albo podaj istniejącego adresata';
    END IF;
    -- Wyznaczenie kwoty rachunku na podstawie rozmiaru
    SET @kwota = CASE
        WHEN p_rozmiar = 'S' THEN 9.5
        WHEN p_rozmiar = 'M' THEN 12.5
        WHEN p_rozmiar = 'L' THEN 19
        WHEN p_rozmiar = 'XL' THEN 28.5
        ELSE 0 -- Domyślna wartość, jeśli rozmiar nie pasuje
    END;
    -- Dodanie nowego rachunku
    INSERT INTO rachunek (data_wystawienia, kwota, forma_platnosci,
status_platnosci, data_opłacenia)
VALUES (NOW(), @kwota, p_forma_platnosci, 'nieopłacona', NULL);
    -- Pobranie ID nowo utworzonego rachunku
    SET @last_rachunek_id = LAST_INSERT_ID();
    -- Dodanie nowej przesyłki
    INSERT INTO przesyłka (waga, rozmiar, ID_nadawcy, ID_adresata,
ID_rachunku)
VALUES (p_waga, p_rozmiar, p_ID_nadawcy, p_ID_adresata,
@last_rachunek_id);
    -- Pobranie ID nowo utworzonej przesyłki
    SET @last_przesylka_id = LAST_INSERT_ID();
    -- Dodanie początkowego stanu przesyłki
    INSERT INTO opis_stanu_przesylki (data_zmiany_stanu,
lokalizacja_paczki, ID_przesyłki, ID_stanu, ID_magazynu)
VALUES (NOW(), 'u nadawcy', @last_przesylka_id, 1, NULL);
END$$
DELIMITER ;
```

Do wywołania procedury należy wprowadzić ID_nadawcy, ID_adresata, wagę, rozmiar i formę płatności, np.: `CALL Dodaj_Nowa_Przesylke(1,2,5.5,'M','przelew')`

Wyzwalacze

- 1) Wyzwalacz, który wywołuje procedurę przypisania kuriera do nadawcy oraz dodaje stan przesyłki: „oczekuje na wysłanie” po opłaceniu przesyłki (po zmianie statusu opłacenia w tabeli *rachunek* na „opłacona”).

```
DELIMITER $$
CREATE TRIGGER Trg_Rachunek_Platnosc
BEFORE UPDATE ON rachunek
FOR EACH ROW
BEGIN
    IF NEW.status_platnosci = 'opłacona' AND OLD.status_platnosci !=
'opłacona' AND NOT EXISTS (
        SELECT 1
        FROM opis_stanu_przesylki
        WHERE ID_przesylki = (SELECT ID_przesylki FROM przesyłka WHERE
ID_rachunku = NEW.ID_rachunku)
        AND ID_stanu = 2 -- Sprawdzamy, czy nie ma wpisu z ID_stanu =
2
    ) THEN
        CALL przypisz_kuriera_do_nadawcy(
            (SELECT kod_pocztowy FROM nadawca WHERE ID_nadawcy
= (SELECT ID_nadawcy FROM przesyłka WHERE ID_rachunku = NEW.ID_rachunku
LIMIT 1)),
            (SELECT ID_przesylki FROM przesyłka WHERE
ID_rachunku = NEW.ID_rachunku LIMIT 1)
        );
        -- Dodanie nowego stanu przesyłki
        INSERT INTO opis_stanu_przesylki (data_zmiany_stanu,
lokalizacja_paczki, ID_przesylki, ID_stanu, ID_magazynu)
VALUES (
        NOW(),
        'u nadawcy',
        (SELECT ID_przesylki FROM przesyłka WHERE ID_rachunku =
NEW.ID_rachunku),
        2,
        NULL
    ); -- Wywołanie procedury przypisz_kuriera_do_nadawcy
    SET NEW.data_oplacenia = NOW(); -- Ustawienie daty opłacenia
na teraz
    END IF;
END$$
DELIMITER ;
```

- 2) Wyzwalacz, który po aktualizacji daty zakończenia realizacji dostawy przez kuriera, który odebrał paczkę od nadawcy, dodaje nowy stan przesyłki „W magazynie”.

```
DELIMITER $$
CREATE TRIGGER
trg_dodaj_stan_5_w_magazynie_po_aktualizacji_daty_zakonczenia
AFTER UPDATE ON realizacja_dostawy
FOR EACH ROW
BEGIN
    -- Sprawdzamy czy kolumna data_zakonczenia zostala zaktualizowana
    IF OLD.data_zakonczenia IS NULL AND NEW.data_zakonczenia IS NOT NULL
    THEN
        -- Sprawdzamy, czy istnieje dokładnie jedna przesyłka dla danego
        ID_przesylki
        IF (SELECT COUNT(*)
            FROM realizacja_dostawy
            WHERE ID_przesylki = NEW.ID_przesylki) = 1 THEN
            -- Dodajemy nowy wiersz do tabeli opis_stanu_przesylki
            INSERT INTO opis_stanu_przesylki (data_zmiany_stanu,
            lokalizacja_paczki, ID_przesylki, ID_stanu, ID_magazynu)
            VALUES (NOW(), 'w magazynie', NEW.ID_przesylki, 5, NULL);
        END IF;
    END IF;
END$$
DELIMITER ;
```

- 3) Wyzwalacz, który wywołuje procedurę przypisania kuriera do adresata po dodaniu stanu przesyłki: „W magazynie docelowym”.

```
DELIMITER $$
CREATE TRIGGER trg_przypisz_kuriera_do_adresata
AFTER INSERT ON opis_stanu_przesylki
FOR EACH ROW
BEGIN
    -- Sprawdzamy czy dodany wiersz ma ID_stanu = 14
    IF NEW.ID_stanu = 14 AND (SELECT COUNT(*) FROM realizacja_dostawy WHERE
    ID_przesylki = NEW.ID_przesylki) = 1 THEN
        CALL przypisz_kuriera_do_adresata(
            (SELECT kod_pocztowy
             FROM adresat
             WHERE ID_adresata = (SELECT ID_adresata
                                 FROM przesyłka
                                 WHERE ID_przesylki = NEW.ID_przesylki
                                 LIMIT 1)),
            NEW.ID_przesylki
        );
    END IF;
END$$
DELIMITER ;
```

- 4) Wyzwalacz, który dodaje stan przesyłki „Dostarczona” po aktualizacji daty zakończenia realizacji dostawy przez kuriera, który dostarczył przesyłkę do adresata.

```
DELIMITER $$
CREATE TRIGGER ustaw_stan_15_po_drugiej_dostawie
AFTER UPDATE ON realizacja_dostawy
FOR EACH ROW
BEGIN
    DECLARE v_count INT;
    DECLARE v_ID_przesylki INT;

    -- Sprawdź, czy data_zakonczenia została ustawiona w najnowszym wpisie
    IF NEW.data_zakonczenia IS NOT NULL THEN
        -- Pobierz ID przesyłki dla tego wpisu
        SET v_ID_przesylki = NEW.ID_przesylki;

        -- Policz ile zakończonych dostaw istnieje dla tej przesyłki
        SELECT COUNT(*) INTO v_count
        FROM realizacja_dostawy
        WHERE ID_przesylki = v_ID_przesylki
        AND data_zakonczenia IS NOT NULL;

        -- Jeśli to druga zakończona dostawa, ustaw stan 15 i zaktualizuj
        opis_stanu_przesylki
        IF v_count = 2 THEN
            INSERT INTO opis_stanu_przesylki (data_zmiany_stanu,
            lokalizacja_paczki, ID_przesylki, ID_stanu, ID_magazynu)
            VALUES (NEW.data_zakonczenia, 'dostarczona', v_ID_przesylki,
            15, NULL);
        END IF;
    END IF;
END$$
DELIMITER ;
```

- 5) Wyzwalacz, który po dodaniu nowego pracownika w tabeli pracownik, dodaje jego ID do tabel kurier, magazynier lub dostawca, gdy ich stanowisko to „Kurier”, „Magazynier” lub „Dostawca”.

```
DELIMITER $$
CREATE TRIGGER trg_dodaj_nowego_pracownika_do_innej_tabeli
AFTER INSERT ON pracownik
FOR EACH ROW
BEGIN
    -- Jeśli stanowisko = 'Kurier', dodaj do tabeli kurier
    IF NEW.stanowisko = 'Kurier' THEN
        INSERT INTO kurier (ID_pracownika, obszar_działania,
kod_pocztowy_od, kod_pocztowy_do)
        -- Ustawiamy defaultowe wartości, trzeba je potem zedytować
        VALUES (NEW.ID_pracownika, "DO ZMIANY", NULL, NULL);
    END IF;

    -- Jeśli stanowisko = 'Magazynier', dodaj do tabeli magazynier
    IF NEW.stanowisko = 'Magazynier' THEN
        INSERT INTO magazynier (ID_pracownika, strefa_zarządzania,
ID_magazynu)
        -- Ustawiamy defaultowe wartości, trzeba je potem zedytować
        VALUES (NEW.ID_pracownika, "DO ZMIANY", 17);
    END IF;

    -- Jeśli stanowisko = 'Dostawca', dodaj do tabeli dostawca
    IF NEW.stanowisko = 'Dostawca' THEN
        INSERT INTO dostawca (ID_pracownika, ID_magazynu_lokalnego,
punkty_docelowe_dostawy, dystans)
        -- Ustawiamy defaultowe wartości, trzeba je potem zedytować
        VALUES (NEW.ID_pracownika, 17, "DO ZMIANY", 0);
    END IF;
END$$
DELIMITER ;
```


Aplikacja

Aplikację wykonałam w pythonie w środowisku PyCharm. Połączenie z bazą odbywa się dzięki `mysql.connector`, a do stworzenia interfejsu graficznego wykorzystałam `tkinter`. Raporty są generowane za pomocą `ReportLab` – biblioteki pythonowej.

Połączenie z bazą

```
def __init__(self, host, port, user, password, database): #nawiązanie
połączenia z bazą
    self.connection = None
    try:
        self.connection = mysql.connector.connect(
            host=host,
            port=port,
            user=user,
            password=password,
            database=database
        )
        if self.connection.is_connected():
            print("Połączono z bazą danych.")
    except Error as e:
        messagebox.showerror("Błąd połączenia", f"Nie udało się połączyć z
bazą danych: {e}")
```

Tworzenie raportu w pythonie – ReportLab:

```
doc = SimpleDocTemplate("lista_pracownikow.pdf", pagesize=letter)
elements = []
```

1 linijka służy do stworzenia dokumentu `lista_pracownikow.pdf`, który zostanie zapisany w katalogu projektu. 2 linijka tworzy listę, która będzie przechowywać elementy raportu.

```
title = Paragraph("Lista pracowników", title_style)
subtitle = Paragraph("według stanowisk", subtitle_style)

elements.append(title)
elements.append(subtitle)
```

Dodajemy elementy, które ma posiadać raport.

```
doc.build(elements)
print("Raport z grupowaniem zapisano jako 'lista_pracownikow.pdf'.")
```

Na sam koniec raport jest budowany i gotowy do pobrania.

Cała aplikacja składa się z dwóch plików: **main.py** oraz **raport.py**. W pliku `main.py` obsługiwany jest cały interfejs graficzny, a w `raport.py` są przechowywane dane raportów, które musiałam stworzyć w samym pythonie (style sekcji + zawartość potrzebna do zbudowania raportu). `Raport.py` służy również do generowania raportów. Połączyłam te pliki w następujący sposób:

main.py (przykład wyboru raportu z wykresem – pobierane są parametry od użytkownika „Data_od” oraz „Data_do”, a następnie przekazywane do „raport.py” za pomocą subprocess. Pozwala to na uruchomienie generowania raportu z wykresem w pliku „raport.py”):

```
if report_type == "Wykres":
    tk.Label(param_window, text="Data początkowa (YYYY-MM-DD):").pack(pady=5)
    start_date_entry = tk.Entry(param_window, width=20)
    start_date_entry.pack(pady=5)

    tk.Label(param_window, text="Data końcowa (YYYY-MM-DD):").pack(pady=5)
    end_date_entry = tk.Entry(param_window, width=20)
    end_date_entry.pack(pady=5)

    def submit_params():
        start_date = start_date_entry.get().strip()
        end_date = end_date_entry.get().strip()
        if not start_date or not end_date:
            messagebox.showerror("Błąd", "Wprowadź obie daty.")
            return
        param_window.destroy()
        subprocess.run(["python", "raport.py", "2", start_date, end_date])
# Wywołanie raportu z wykresem
        messagebox.showinfo("Sukces", "Raport został zapisany jako \\liczba_dostaw.pdf\\".")

    tk.Button(param_window, text="Wygeneruj raport",
command=submit_params).pack(pady=10)
```

raport.py (obsługa parametrów oraz wyboru raportu - przekazanych z pliku „main.py”):

```
def main():
    if len(sys.argv) < 2:
        print("Brak argumentu określającego typ raportu. Wybierz 1, 2 lub 3.")
        return

    choice = sys.argv[1] # Typ raportu
    if choice == "1": # Raport z grupowaniem
        generate_grouped_report()
    elif choice == "2": # Raport z wykresem
        if len(sys.argv) < 4: # Sprawdź, czy są przekazane daty
            print("Brak wymaganych parametrów (start_date, end_date) dla raportu z wykresem.")
            return
        start_date = sys.argv[2]
        end_date = sys.argv[3]
        generate_chart_report(start_date, end_date)
    elif choice == "3": # Raport w formie formularza
        if len(sys.argv) < 3: # Sprawdź, czy jest przekazane ID przesyłki
            print("Brak wymaganego parametru (id_przesylki) dla raportu w formie formularza.")
            return
        id_przesylki = sys.argv[2]
        generate_form_report(id_przesylki)
    else:
        print("Nieprawidłowy wybór. Wybierz 1, 2 lub 3.")
```

Okno wyboru raportu w pliku main.py:

```
#główne okno wyboru raportu
report_window = tk.Toplevel(self.root)
report_window.title("Wybierz typ raportu")
report_window.geometry("400x200")

tk.Label(report_window, text="Wybierz rodzaj raportu:").pack(pady=10)

tk.Button(report_window, text="Lista pracowników na danych stanowiskach",
          command=lambda: [report_window.destroy(),
                           generate_grouped_report()]).pack(pady=5)
tk.Button(report_window, text="Realizacja dostaw przez kurierów w
określonym czasie",
          command=lambda: [report_window.destroy(),
                           open_param_window("Wykres")]).pack(pady=5)
tk.Button(report_window, text="Szczegóły przesyłki",
          command=lambda: [report_window.destroy(),
                           open_param_window("Formularz")]).pack(pady=5)
```

Działanie aplikacji:

Wyświetlanie danych:

Zarządzanie danymi w bazie

ID_magazyniera	strefa_zarzadzania	ID_magazynu
5	strefa_A	1
7	strefa_B	2
9	strefa_C	3
11	strefa_D	4
13	strefa_E	5
15	strefa_F	6
17	strefa_G	7
23	strefa_H	8
25	strefa_I	9
27	strefa_J	10
29	strefa_K	11
31	strefa_L	12
33	strefa_M	13
35	strefa_N	14
37	strefa_O	17
39	strefa_P	17

Powiązane dane dla: ID_magazyniera → pracownik (ID_pracownika)

ID_pracownika	imię	nazwisko	numer_tel	stanowisko	pensja	ID_przełożonego
1	Adam	Nowak	123456789	Dyrektor	12000.0	None
2	Bartek	Zielinski	987654321	Kierownik	5200.0	1
3	Cezary	Adamski	564738912	Kierownik	5200.0	1
4	Dawid	Wojcik	987456321	Kurier	3000.0	2
5	Ewa	Sikora	948456789	Magazynier	3200.0	3
6	Filip	Kowalczyk	786431987	Kurier	3100.0	2
7	Grzegorz	Zawadzki	984761234	Magazynier	3200.0	3
8	Hanna	Mazur	987654321	Kurier	3100.0	2

Powiązane dane dla: ID_magazynu → magazyn (ID_magazynu)

ID_magazynu	miasto	ulica	pojemnosc	kod_pocztowy	wojewodztwo
1	Warszawa	Magazynowa 1	1000	00-030	Mazowieckie
2	Kraków	Magazynowa 2	800	30-004	Małopolskie
3	Gdańsk	Magazynowa 3	1200	80-009	Pomorskie
4	Wrocław	Magazynowa 4	900	50-010	Dolnośląskie
5	Poznań	Magazynowa 5	1100	60-059	Wielkopolskie
6	Szczecin	Magazynowa 6	950	70-002	Zachodnio-Pomorskie
7	Lublin	Magazynowa 7	1300	20-033	Lubelskie
8	Bydgoszcz	Magazynowa 8	1000	85-070	Kujawsko-Pomorskie

Dodawanie danych:

Dodaj rekord do tabeli: kurier

ID_kuriera:

obszar_dzialania:

kod_pocztowy_od:

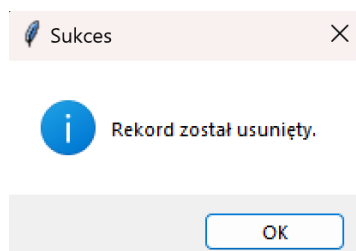
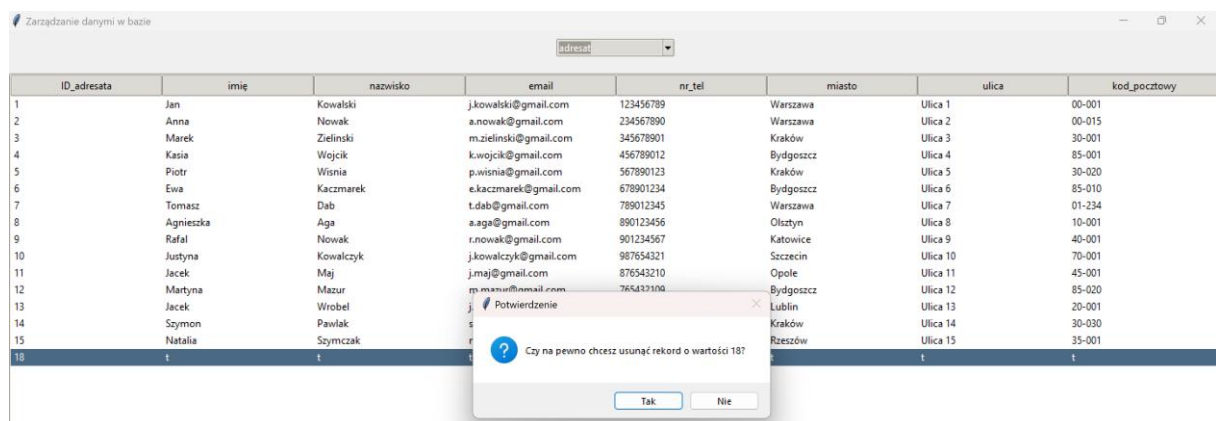
kod_pocztowy_do:

Zapisz rekord

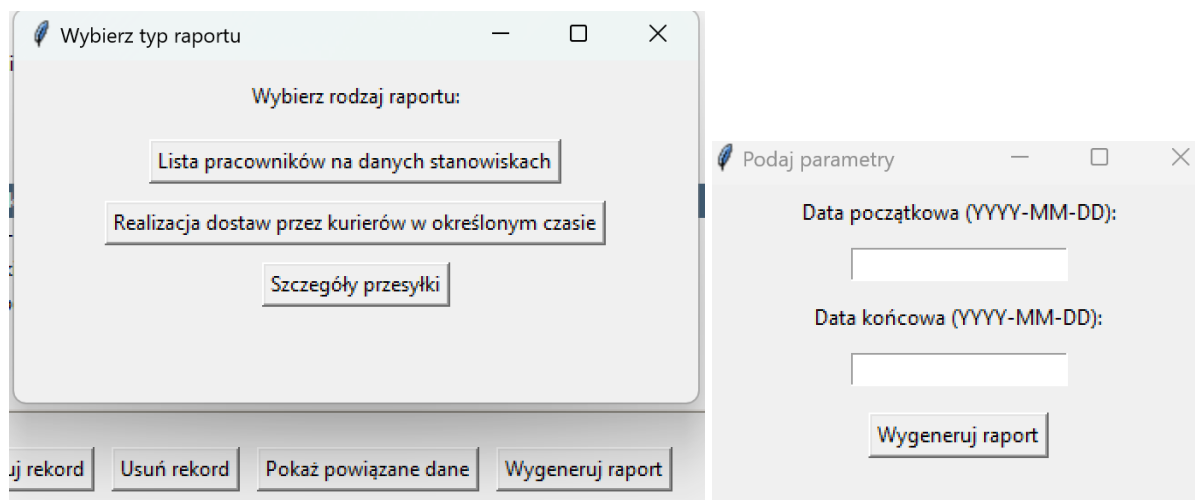
Powiązane dane dla: ID_kuriera → pracownik (ID_pracownika)

ID_pracownika	imię	nazwisko	numer_tel	stanowisko	pensja	ID_przełożonego
1	Adam	Nowak	123456789	Dyrektor	12000.0	None
2	Bartek	Zielinski	987654321	Kierownik	5200.0	1
3	Cezary	Adamski	564738912	Kierownik	5200.0	1
4	Dawid	Wojcik	987456321	Kurier	3000.0	2
5	Ewa	Sikora	948456789	Magazynier	3200.0	3
6	Filip	Kowalczyk	786431987	Kurier	3100.0	2
7	Grzegorz	Zawadzki	984761234	Magazynier	3200.0	3
8	Hanna	Mazur	987654321	Kurier	3100.0	2

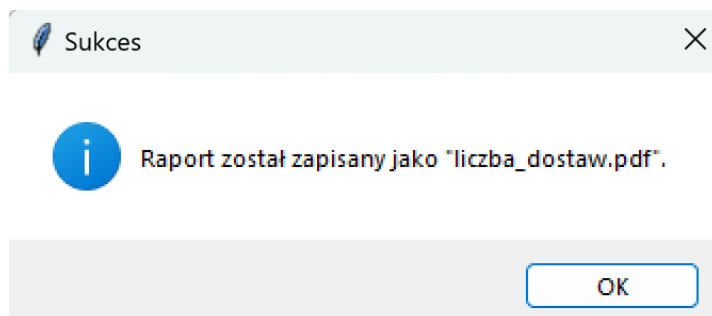
Usuwanie danych:



Generowanie raportu:



Pomyślne wygenerowanie raportu:



Szablony raportów

Raport 1

Podawane parametry: ID_przesyłki

Przesyłka	
Zestawienie informacji o przesyłce	
Data: 2025-01-05	
Szczegóły przesyłki	
ID przesyłki: 8	
Waga: 5.0 kg	
Rozmiar: XL	
Stan przesyłki	
Aktualny stan: Dostarczona	
Lokalizacja: dostarczona	
Data ostatniej zmiany stanu: 2024-11-05 11:04:14	
Nadawca	
Imię i nazwisko: Henryk Henrykowski	
Adres: Ulica H, Lublin, 20-221	
Numer telefonu: 890890890	
Adresat	
Imię i nazwisko: Agnieszka Aga	
Adres: Ulica 8, Olsztyn, 10-001	
Numer telefonu: 890123456	
Rachunek	
Status płatności: opłacona	
Kwota: 28.5 PLN	
Data wystawienia: 2024-11-03 19:00:00	

Raport 2

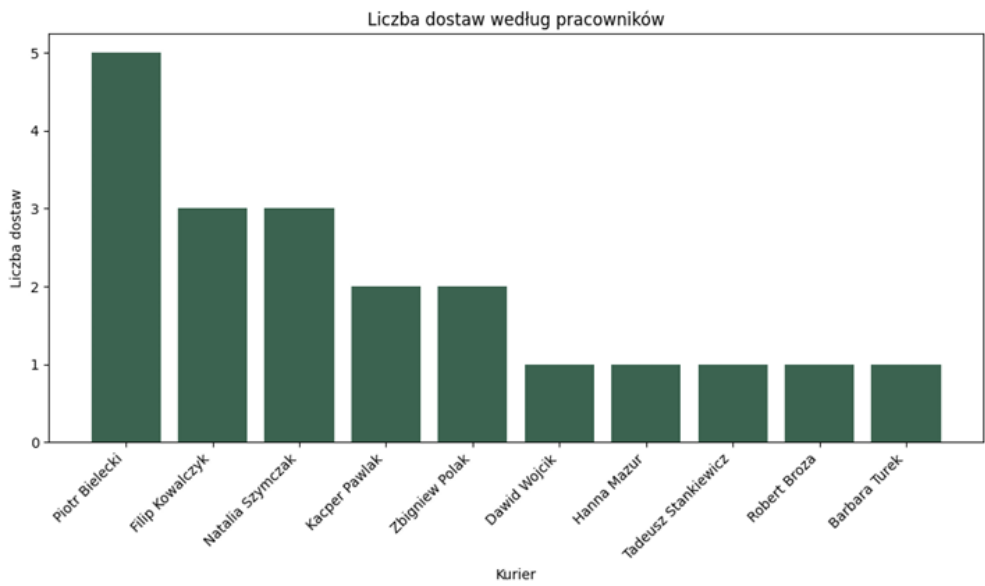
Podawane parametry: Data_od, Data_do

Liczba zrealizowanych dostaw przez kurierów

w okresie: 2024-01-01 - 2024-12-31

Date: 2025-01-05

Imię	Nazwisko	Pensja	Liczba Dostaw
Piotr	Bielecki	3050.0	5
Filip	Kowalczyk	3100.0	3
Natalia	Szymczak	3000.0	3
Kacper	Pawlak	3100.0	2
Zbigniew	Polak	3200.0	2
Dawid	Wojcik	3000.0	1
Hanna	Mazur	3100.0	1
Tadeusz	Stankiewicz	3150.0	1
Robert	Broza	3000.0	1
Barbara	Turek	3100.0	1



Raport 3:

Lista pracowników według stanowisk

Data: 2025-01-05

Stanowisko: Dostawca

Imię	Nazwisko	Numer telefonu	Pensja
Izabela	Baran	123456987	3100.0
Ryszard	Bartosz	654123456	3150.0
Jacek	Dąbrowski	456123789	3000.0
Nina	Dębowska	456789987	3000.0
Michał	Gajewski	321654789	3150.0
Daria	Gorska	789321654	3100.0
Edward	Jurek	321789654	3000.0
Alicja	Kowalska	321789987	3100.0
Gabriela	Król	654321123	3050.0
Paulina	Kucharska	321789123	3000.0
Sara	Lisowska	123654987	3050.0
Bartosz	Malinowski	789456123	3000.0
Piotr	Mazowiecki	789123987	3100.0
Krzysztof	Nowak	123789456	3100.0
Cezary	Nowicki	654123789	3150.0
Leon	Ostrowski	321654321	3050.0
Oliwier	Pietrzak	654987123	3200.0
Katarzyna	Polanska	789123654	3150.0
Franciszek	Rataj	456987321	3100.0
Norbert	Wieczorek	654321456	3000.0
Olaf	Wróbel	987321123	3100.0
Lidia	Zajac	789654987	3050.0
Hubert	Zakrzewski	789654789	3150.0
Marcelina	Zięba	456789123	3200.0

Stanowisko: Dyrektor

Imię	Nazwisko	Numer telefonu	Pensja
Adam	Nowak	123456789	12000.0